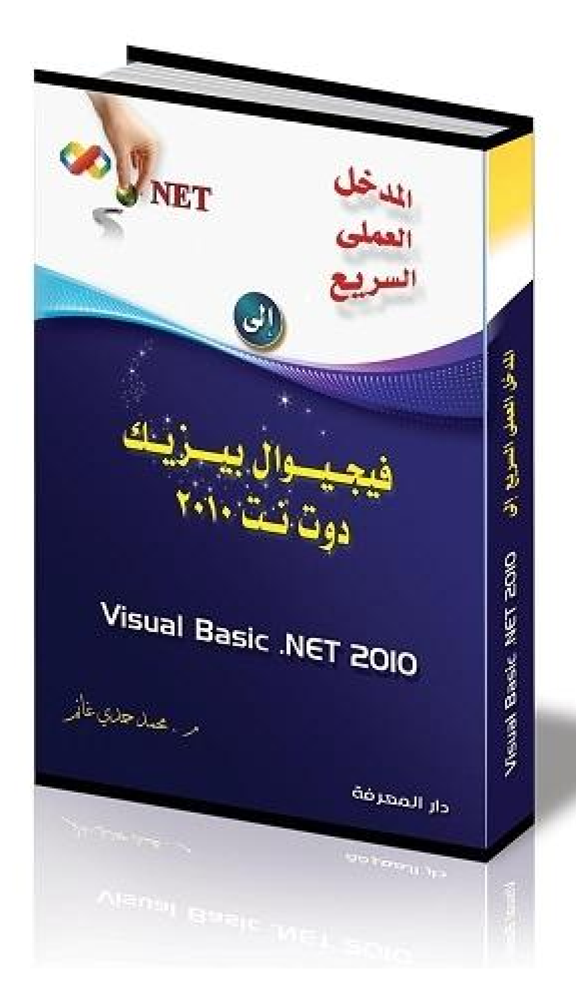
إنشاء قواعد بيانات SQL Server 2008 باستخدام LinQ To SQL

فصل من كتاب: **المدخل العملي السريع إلى فيجيوال بيزيك دوت نت ٢٠١٠**



بقلم: م. محمد حمدي غانم
http://mhmdhmdy.blogspot.com

طباعة ونشر: مكتبة دار المعرفة

٤ شارع السرايات ــ أمام هندسة عين شمس ــ بالقرب من ميدان عبده باشا ــ العباسية ــ القاهرة، هاتف **٠٢/٢٦٨٤٤٠٤٣**

مقـدمــة الكــتاب

بسم الله والصلاة والسلام على رسول الله وآله وصحبه، وبعد:

هذا الكتاب موجه إلى القارئ الشغوف، الذي لا يعرف ما هي فيجيوال بيزيــك دوت نت، أو ما هي البرمجة عموما، ويريد أن يأخذ جولة سريعة مختصــرة، ولكن في نفس الوقت وافية، وكافية لكي يقرر إن كانت البرمجة هــي المجــال الذي يناسبه، وفيجيوال بيزيك هي لغة البرمجة التي يستطيع الاعتماد عليها، أم لا.

وهو موجه أيضا إلى الطالب أو الدارس الذي يريد أن يلمّ فــي أســرع وقــت بمهارات فيجيوال بيزيك الأساسية، التي تؤهله لدراسة أحد المناهج، أو تساعده في إنجاز مشروع التخرج أو كتابة برنامج مساند لرسالته البحثية.

وهو موجه كذلك إلى الموظف أو المحاسب، الذي سمع عــن قــدرة فيجيــوال بيزيك على تطوير تطبيقات تجارية قوية، فانتابه الفضول للتعرف علــى هــذه اللغة، وشعر بالرغبة في تجربتها بنفسه، لإنشاء برنامج سريع ينظم به عملــه، ويطبق فيه كل الأفكار التي لم تستطع البرامج الجاهزة تقديمها له.. هذا الكتاب يعطيه مثالا جيدا لمشروع يعمله كيفيه إنشاء قواعد البيانات واســتخدامها فــي حفظ البيانات والبحث فيها وعرضها بأشكال مختلفة.

وهو موجه بوجه عام للمبتدئين في البرمجة، الذين يريـــدون مشـــاريع كبيـــرة مشروحة بالكامل، يستطيعون الاستفادة بأجزاء منها أثناء بـــرمجتهم للمشـــاريع المطلوبة منهم.

باختصار: هذا الكتاب مفيد لكل الذين ملوا من قراءة الأساســيات والتعليمـــات والأمثلة الصغيرة، ويريدون أن يتعلموا كيف يطورون مشروعا عمليا، يـــربط الأجزاء التي تعلموها بعضها ببعض، ويمنحهم الكثير من الخبرات العملية.

منهج الكتاب:

لكي يحقق هذا الكتاب كل هذه الأهداف، تم تصميمه ليقفز بقارئه مباشرة إلـــى منتصف ورشة العمل، أثناء تطوير مشروع متكامل اسمه المذكرة الذكية، بدون أي مقدمات أو شروح نظرية لأساسيات البرمجة أو لغة فيجيوال بيزيك، فبـــدلا من هذا، يريك الكتاب خطوات العمل على الهواء مباشرة، وكلما تطلب الأمـــر شرح جزء من لغة فيجيوال بيزيك، يتم شرحه باختصـــار واف، وعلـــى قـــدر الحاجة.

ويقدم الكتاب تصميمين مختلفين للمذكرة الذكية، أحـــدهما برنـــامج شخصـــي لمستخدم واحد يحفظ البيانات في ملف على الجهاز، والآخـــر برنـــامج متعـــدد المستخدمين، يتيح لأكثر من مستخدم التعامل معه في نفس الوقت، مـــن خـــلال قواعد بيانات "سيكويل سيرفر" SQL Server.

وهكذا، وبنهاية هذا الكتاب، يكون القارئ قد أخذ نظرة واسعة علـــى فيجـــوال بيزيك، وتعلم ما يلي:

- أساسيات لغة البرمجة، كتعريف المتغيرات، وحلقات التكرار Loops، وجمل الشرط، واستخدام المجموعات Collections، وتعريف الدوال Functions والسجلات Structures والفئات Classes.
- برمجة نماذج الويندوز، وتصميم النماذج وواجهة الوثائق المتعددة، واستخدام الأدوات الأساسية، كالأزرار ومربعات النصوص ومربعات القوائم ListBoxes، وأزرار التحويل RadioButtons ومربعات الاختيار CheckBoxes، واللوحة Panel ومربع التجميع GroupBox، وأداة مزود الخطأ ErrorProvider، وأداة تلميحات الشاشة ToolTip، والقوائم الرئيسية Menus.
- إنشاء أداة خاصة Custom Control، لعرض المواعيد بصورة مرئية واضحة وسهلة، واستخدامها في تصميم المذكرة الذكية.
- إجادة الكثير من المواضيع المتقدمة، كالنقل المتسلسل للبيانات Serialization وكتابة استعلامات LinQ، وبرمجة قواعد بيانات SQL Server من خلال النموذج التصوري Conceptual Model الذي تقدمه لنا تقنية LinQ To SQL، وربط البيانات Data Binding، وعرضها في الأدوات المتقدمة مثل جدول العرض DataGridView، وإنشاء أعمدة من أنواع خاصة Custom DataGridView Columns، لعرض البيانات في جدول العرض بالشكل الذي يناسبنا.

وغير ذلك الكثير!

أعرف أنك مندهش، وأنك ستقلب الكتاب بين يديك وتقول غير مصدق:

- كيف يمكن لكتاب صغير الحجم بهذا الشكل، أن يقدم مشروعين كبيرين ويشرحهما خطوة بخطوة، ويتناول في نفس الوقت كل هذه المواضيع؟!

ولكنها الحقيقية يا صديقي بدون أي مبالغة، فبين يديك كتاب غير تقليدي، ستستمتع بكل جملة تقرؤها فيه، لأنه مصمم أساسا ليجعلك تعرف لماذا يعشق الآخرون البرمجة.

دعنا لا نضيع المزيد الوقت في مقدمات، فلدينا مشروعان شيقان يناديان أن هلموا!

<u>متطلبات أساسية في قارئ الكتاب</u>:

لا يشترط هذا الكتاب في قارئه أي معرفة مسبقة بلغة فيجيوال بيزيك، ولا بأي لغة برمجة على الإطلاق.

وليس شرطا أن يكون طالبا في كلية الهندسة أو الحاسبات والمعلومات.. فالمشروعان الموجودان في هذا الكتاب لا يتطلبان أي خلفية معرفية في الرياضيات أو المحاسبة أو أي علم متخصص.. بل يمكن حتى لطالب المرحلة الثانوية قراءته وتنفيذ ما فيه بدون أي تعقيد.

كل المطلوب من قارئ هذا الكتاب، هو ما يلي:

- أن يجيد أساسيات التعامل مع الحاسوب ونظام تشغيل الويندوز.
- أن يعرف الحد الأدنى من اللغة الإنجليزية، لأن أوامر لغة البرمجة تكتب بها.. وهي إنجليزية بسيطة، لا تزيد عن مستوى طالب الصف الثالث الإعدادي، فلغة البرمجة لا تهتم بقواعد النحو والصرف، وقاموس الكلمات

المستخدم فيها محدود للغاية، ويكفي أن يمتلك المرء القدرة علـــى قـــراءة الكلمات الإنجليزية لكي يألفها بسرعة.. وعموما، أنا أترجم في هذا الكتاب كل مصطلح برمجي وكل أمر من أوامر اللغة، حيث أكتب الكلمة العربية بجوار الكلمة الإنجليزية.

- أن يعدّ لغة البرمجة على جهازه، وقد شرحنا خطوات الإعداد بالتفصـــيل في بداية هذا الكتاب.

وعليّ هنا أن أقدم نصيحة للقارئ حتى يخرج بأقصى استفادة ممكنة من الكتاب بإذن الله:

يجب أن تنفذ كل حرف نشرحه في هذا الكتاب بيدك عمليا.. فهذا الكتاب لـــيس للتسلية في الحافلة أو القطار، ولا يمكنك قراءته وأنت مضطجع في فراشـــك.. يجب أن تكون بجوار الحاسوب وأنت تقرأ هذا الكتاب، وأن تنفذ كل خطوة في المشروع بعد أن تفهم شرحها جيدا، لأنك لو فقدت تسلسل الفهـــم، أو اعتمـــدت على قراءة الكتاب كأنه جريدة ونسيت في منتصفه ما قرأته في أوله، بسبب عدم ممارستك له عمليا وعدم كتابة الكود بيدك، فستجد نفسك عاجزا عن المواصـــلة أو الاستفادة.

ولا يهم أن تنتهي من قراءة الكتاب بسرعة، بل المهم ألا تنتقل من فقـــرة إلـــى أخرى إلا بعد فهمها فهما جيدا، وتطبقها بنفسك في فيجيـــوال بيزيـــك، فـــتعلّم البرمجة أشبه بصعود سلم، كل درجة فيه تفضي إلى الدرجة التالية لهـــا، ولـــو فقدت بضع درجات فستسقط سريعا ولن ترقى إلى مبتغاك.

وأنصحك أخيرا بأن تخصص وقتا للكتاب يوميا حتى لو كان صـــغيرا، فهجـــر الكتاب لفترة طويلة قد يجعلك تنسى ما سبق أن تعلمته من أساسيات البرمجـــة،

كما أنك قد تنسى أسماء النماذج ووظائفها والمنطق العـــام الـــذي نتبعـــه فـــي المشروع، مما يصعب عليك متابعته دون مراجعة ما سبق.

هذه فقط هي كل المؤهلات المطلوبة منك، وكما ترى، فهي مـــؤهلات يســـهل توفرها في كل من يحب ما يتعلمه ولديـــه الحمـــاس الكـــافي والصـــبر علـــى المواصلة.

والله ولي التوفيق

لمراسلة الكاتب والتواصل معه:

- البريد الالكتروني:

msvbnet@hotmail.com

- مدونة م. محمد حمدي غانم:

http://mhmdhmdy.blogspot.com

مشروع المذكرة الذكية
باستخدام قواعد البيانات

قاعدة البيانات Database:
هي ملف أو مجموعة من الملفات، تستخدم لتخزين البيانات.. وتتميـز قاعدة البيانات عن الملفات العادية، بوجود برامج خاصة للتعامل معها، مما يجعل حفظ البيانات وقراءتها والبحث فيها أمـورا أكثـر سـهولةً وتنظيمًا، وخالية من الأخطاء.. هذه البرامج تسمّى "أنظمة إدارة قواعد البيانات":   Database Management Systems (DBMS)   وتحتوي قاعدة البيانات على أنواع مختلفة من العناصر، منها الجـداول Tables والعلاقات Relations.. حيث توضع البيانات في الجـداول، التي يمكن أن ترتبط بعلاقات فيما بينها.. ويتكون كل جدول من: - أعمدة Columns لكل منها اسم محدد ونوع بيانـات محـدد.. ويكون عدد أعمدة الجدول ثابتا.. ويطلق على العمود Column أيضا اسم الحقل Field.   - وصفوف Rows توضع فيها البيانات.. ويمكنك إضافة مئات أو آلاف أو ملايين الصفوف إلى الجدول على حسـب الحاجـة.. ويطلق على الصف Row أيضا اسم السجل Record.

سيكويل سيرفر ٢٠٠٨:

هناك أنواع كثيرة من قواعد البيانات، مثل قواعد بيانات Access وقواعد بيانات SQL Server وهما من إنتاج ميكروسوفت.. وسنستخدم هنا قواعد بيانات SQL Server، لأن فيجيوال بيزيك دوت نت تقدم لها دعما كاملا، يجعل التعامل معها أسهل وأسرع، كما أن سيكويل سيرفر يتيح لنا تشغيل قاعدة البيانات على خادم Server، مما يجعل كل مستخدمي البرنامج يشتركون في التعامل مع نفس قاعدة البيانات.

ويجب أن يكون سيكويل سيرفر معدا على جهازك لكي يمكننا التعامل مع قواعد البيانات الخاصة به.. وإذا لم يكن هذا البرنامج لديك، فيمكنك استخدام نسخة خفيفة منه اسمها:

SQL Server 2008 Management Studio Express

أو ما يمكن أن نسميه بالعربية مدير سيكويل سيرفر ٢٠٠٨.. ويمكنك الحصول عليه مجانا من صفحة تنزيل البرامج من موقع ميكروسوفت:

http://www.microsoft.com/downloads

في أعلى هذه الصفحة ستجد مربع البحث.. اكتب فيه النص:

SQL Server 2008 Management Studio

واضغط زر البحث (الذي تمثله أيقونة العدسة).. ستجد في نتائج البحث رابط البرنامج الذي تبحث عنه.. اضغطه لعرض صفحة تنزيله، واضغط زر تنزيل النسخة المناسبة لنظام التشغيل الموجود على جهازك.

إعداد Management Studio Express:

قبل أن تبدأ إعداد هذا البرنامج على جهازك، عليّ أن أنبهك إلى ضرورة التأكد من أنك قد اخترت إعداد العنصر:

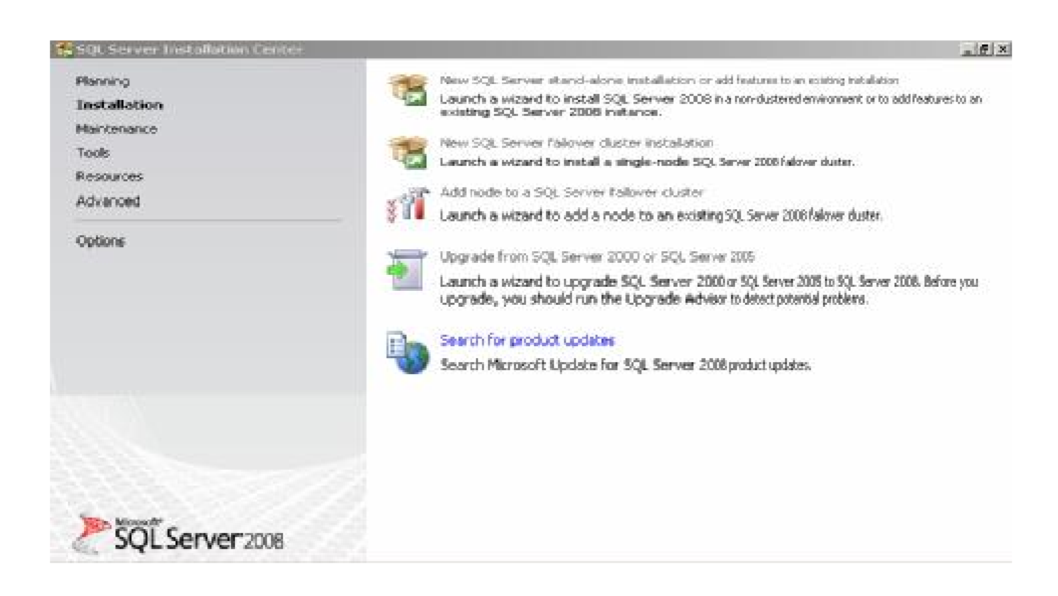
Microsoft SQL Server 2008 Express Edition

أثناء إعداد فيجيوال ستديو دوت نت ٢٠١٠، وإلا فإن عليك إعادة تشغيل ملف إعداد دوت نت، واختيار Add or Remove Features، ومن ثم وضع علامة الاختيار أمام هذا العنصر، حتى تستطيع فيجوال ستديو التعامل مع سيكويل سيرفر ٢٠٠٨.

والآن، اضغط برنامج الإعداد الخاص بـ:

SQL Server 2008 Management Studio Express

وانتظر حتى تظهر الشاشة الموضحة في الصورة:



من الهامش الأيسر، اضغط الاختيار الثاني Install، حيث سيعرض لك الجزء الأيمن من النافذة عدة اختيارات.. إذا لم تكن قد أعددت إصـــدارا ســـابقا مـــن سيكويل سيرفر على جهازك، فاضغط الاختيار الأول.. سيتم أولا التأكـــد مـــن وجود المتطلبات اللازمة للإعداد على جهازك، فـــإن كانـــت كلهـــا موجـــودة، فاضغط OK وواصل باقي خيارات الإعداد.. وتأكد دائما مـــن ضـــغط الـــزر Select All في كل صفحة تعرض العناصر المتاحة للإعداد، لتضمن إعـــداد كافة مكونات البرنامج.

ملحوظة:
قد يحدث خطأ في برنامج الإعداد، حيث ستخبرك الرسالة أنها تحتـــاج واحدا أو أكثر من البرامج التالية: Windows Installer 4.5 WS-Management v1.1 Windows PowerShell V2 (CTP3) .NET Framework 3.5 SP1 ويمكنك الحصول على أي برنامج منها بالبحث عن اسمه فـــي صـــفحة تنزيل البرامج من موقع ميكروسوفت كما شرحنا أعلاه، وإعداده علـــى جهازك إذا تطلب الأمر ذلك.

تصميم قاعدة البيانات:

ابدأ مشروعا جديدا اسمه SmartNotebookDB، واحفظـــه فـــي الموضـــع المناسب على جهازك.

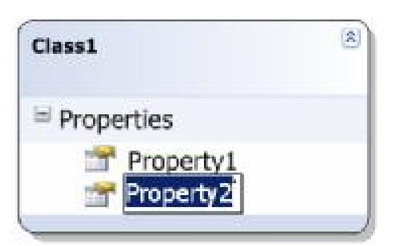
الآن، وقبل أن نصمم نماذج هذا المشروع، علينا أن نصمم قاعدة البيانات أولا.. دائما ما تكون هذه هي أول خطوة في مشاريع قواعد البيانات.

لفعل هذا، افتح القائمة الرئيسية Project واضغط الأمـــر Add New Item، وفي نافذة إضافة عنصر اختر النوع Database من القائمة اليسرى، واختـــر العنصر LinQ To SQL Classes من القائمة اليمنى.. اكتب في خانة الاســـم SmartNoteBookDB واضغط زر الموافقة.. سيؤدي هذا إلى إضافة عنصر جديد إلى متصفح المشاريع اســـمه SmartNoteBookDB.dbml، وســـيتم عرض محتوياته في نافذة تسمى مصمم الكائنات.. هذا العنصر يسمى محتـــوى البيانات DataContext ونظرا لأنه ما زال فارغا، فستظهر النافذة فارغة.

لو فتحت صندوق الأدوات الآن، فستجد فيه أدوات جديدة غير التي نســـتخدمها في تصميم النماذج.. هذه الأدوات تستخدم لتصميم عناصر محتوى البيانات.

محتوى البيانات Data Context:
يسمى أيضا بالنموذج التصوري Conceptual Model، وهو طبقـــة تفصل بيننا وبين قاعدة البيانات، وفيها يتم إنشاء مجموعة من الفئـــات، لتمثل كل منها نوع صفوف أحد الجداول.. وتحتوي كـــل فئـــة علـــى خصائص تمثل كل خاصية منها أحد أعمدة الجدول.. كما يمكن إنشـــاء ارتباط بين فئتين ليمثل العلاقة بينهما.. إذن: الفئة Class ---> صف Row في أحد الجداول الخاصية Property ---> عمود Column في هذا الجدل الارتباط Association ---> علاقة Relation بين جدولين

من صندوق الأدوات اسحب العنصر Class وأسقطه على مصمم الكائنـــات.. سيؤدي هذا إلى إضافة فئة جديدة إلـــى المصـــمم، وسيكون اسمها الافتراضي Class1 كما تبدو في الصورة المجاورة.



ولتغيير اسم الفئة انقره مرتين بالفأرة.. سيؤدي هذا إلى عرض الاسم في مربع نص حتى يمكنك تغييره.. اكتـــب الاســـم Friend واضغط Enter.

ولإضافة خاصية إلى الفئة، اضغطها بزر الفأرة الأيمن، ومن القائمة الموضعية اضغط Add، ومن القائمة الفرعية اضغط الأمر Property.. سيظهر عنـــوان جديد داخـــل المســـتطيل الممثـــل للفئـــة اســـمه Properties، وتحته ســـتظهر خاصـــية جديـــدة، وسيكون اسمها الافتراضي Property1.. ولتغيير هذا الاسم انقره مرتين بالفأرة، واكتب الاسم ID ثم اضغط Enter.



لاحظ أنك لـــو ضـــغطت Enter مـــرة أخـــرى فستضاف خاصية جديدة فـــي وضـــع التحريـــر مباشرة.. هذا تسهيل يجعل إضافة الخصائص أسرع، حيث تكتب الاسم وتضغط Enter مرتين وتكتب الاسم التالي وتضغط Enter مرتين.. وهكذا.

كما أنك تستطيع تغيير اسم الخاصية بضغطها مرة بالفأرة لتحديدها، والكتابة من لوحة المفاتيح مباشرة، حيث سيؤدي هذا إلى الانتقال إلـــى وضـــع التحريـــر..
استخدم هذه الطريقة لإضافة الخصائص التالية إلى الفئة Friend:
ID، Name، Address، BirthDate، Company، Job، Notes.
يجب أن يبدو مخطط الفئة Friend كما في الصورة.
لاحظ أن كل الخصائص التي أضفتها تم جعلها من النوع String تلقائيـــا مـــع منحها خصائص افتراضية.. لهذا علينا تعديل خصائص كل خاصية على حدة..
لفعل هذا، حدد الخاصية في مخطط الفئة بضغطها مرة واحدة، ثم اضـــغط F4 لعرض نافذة الخصائص، واستخدمها لتعديل خصائص الخاصية.

بالنسبة للعمود ID:

هذا العمود سيعمل كمفتاح أساسي Primary Key لجدول الأصدقاء.. المفتـــاح الأساسي هو عمود يحتوي على قيمة متفردة لكل صف فـــي الجـــدول.. وفـــي الغالب يحتوي هذا العمود على أرقام يتم توليدها آليا، بحيث يأخذ كـــل صـــف رقما خاصا به.. فنقول مثلا إن أحمد رقمه ٥ في جدول الأصـــدقاء، ومحمـــود رقمه ٦٠.. وهكذا.
ولجعل العمود ID مفتاحا أساسيا، يجب أن تضبط قيم خصائصـــه كمـــا هـــو موضح في الجدول التالي:

الخاصية	القيمة	ملاحظات
مفتاح أساسي Primary Key	True	سيظهر رمز المفتاح بجوار العمود ID في مخطط الفئة Person.
قيمة مولدة تلقائيا Auto Generated Value	True	هذه الخاصية تطلب من قاعدة البيانات توليد قيم المفتاح الأساسي تلقائيا.
للقراءة فقط Read Only	True	سيكون المفتاح الأساسي للقراءة فقط، حتى لا نسمح بتغيير قيمته بعد توليده.
تزامن تلقائي Auto-Sync	عند الإضافة OnInsert	ستولد قاعدة البيانات المفتاح الأساسي عندما يتم حفظ سجل جديد في قاعدة البيانات، بينما لن نستطيع نحن تغيير قيمته لأننا جعلناه للقراءة فقط.. لهذا يجب أن نطلب من قاعدة البيانات أن ترسل إلينا قيمة المفتاح الأساسي الذي قامت بتوليده، لتوضع في الخاصية ID تلقائيا، وبذلك يكون برنامجنا متزامنا مع قاعدة البيانات.

لاحظ أن أي مفتاح أساسي تنشئه، يجب أن تضبط خصائصه كما في الجـــدول السابق.
بقي أن نصحح نوع البيانات التي ستوضع في العمود ID.. لـــو نظـــرت إلـــى الخاصية Type، فستجد أنها من النوع String.. غيرها إلى Integer ليكـــون المفتاح الأساسي عددا صحيحا.

بالنسبة للعمود Name:
لن نغير شيئا في خصائصه، لأنه بالفعل يتعامل مع نصوص عادية.

وبالنسبة للأعمدة Address، Company، Job، Notes:
فهي أيضا أعمدة نصية، ولكننا سنغير فيها شيئا واحدا.. فنحن نريد أن نســـمح لهذه الأعمدة بأن تكون فارغة، فربما لا يعرف المستخدم الشركة التي يعمل فيها صديقه، أو لا توجد لديه ملحوظات.. إلخ.
تتحكم في هذا خاصية القابلية للانعدام Nullable، وتكون القيمة الافتراضية لها False لكي لا يتم ترك أي خانة في العمود فارغة.. غير هذه القيمة إلى True في كل من الأعمدة Address، Company، Job، Notes.

وأخيرا بالنسبة للعمود BirthDate:
استخدم الخاصية Type لتغيير نوع هذا العمود إلى Date.. وسنســـمح أيضـــا بترك هذا العمود فارغا، لهذا ضع True في الخاصية Nullable.
لا تنسَ ضغط Ctrl+S لحفظ عملك بين فترة وأخرى.

ربط محتوى البيانات بقاعدة البيانات:

تتبقى خاصية هامة، يجب أن نضع قيمتها، وهي خاصـــية المصـــدر Source، وهي تعني العنصر الأصلي في قاعدة البيانات، الذي ســيرتبط بـــه العنصـــر الموجود في محتوى البيانات.. دعنا نتفق على هذين المبدأين:

- كل فئة في محتوى البيانات، سيكون مصدرها جدول اسمه هو لفظ الجمع لاسم الفئة.. على سبيل المثال: الفئة Person سيكون مصدرها جـــدول الأشخاص Persons.. السبب في هذا أن الفئـــة Person تمثـــل صـــفا واحدا في الجدول Persons.. لفعل هذا حدد مخطـــط الفئـــة Person بضغط أي جزء منها خارج جزء الخصائص، واضغط F4 لعرض نافذة الخصائص، واكتب في الخاصية Source الكلمة Persons.
- كل خاصية في الفئة، سيكون مصدرها العمود الذي يحمل نفس اســـمها في الجدول.. لاحظ أن من الممكن أن نجعل كلا الاسمين مختلفـــين دون أن يؤثر هذا على عمل البرنامج.. لكننا حاليا سنجعل للخصائص نفـــس أسماء الأعمدة.. حدد كل خاصية في الفئة Person وضع في الخاصية Source التابعة لها نفس اسمها.

فئة الهاتف Phone:

بنفس الطريقة السابقة، أضف فئة جديدة إلى محتوى البيانات، واجعـــل اســـمها Phone، واستخدم الخاصية Source لربطها بالجدول Phones فـــي قاعـــدة البيانات.. هذا الجدول وظيفته أن نضع فيه كل أرقام هواتف الأشخاص، حيـــث

سنربط رقم الهاتف بصاحبه من خلال عمود اسمه PersonID، سنضع فيه رقم الشخص.. يسمى هذا العمود بالمفتاح الثانوي أو المفتــاح الغريــب Foreign Key، لأنه يحمل قيمة المفتاح الأساسي لجدول آخر.

ولكن: لماذا لم نضع ثلاثة أعمدة في جدول الأشخاص لتحمل هــاتف المنــزل والعمل والمحمول؟.. لماذا أنشأنا جدولا خاصا بهذا الأمر؟

في الحقيقة، هناك ميزتان في هذا:

١- إذا لم يكن للشخص أي هواتف، فلن نترك ثلاث خانــات فــي جــدول الأشخاص فارغة.. بل كل ما سنفعله، هو ألا نضيف أي سجل خاص به في جدول الهواتف.

٢- إذا كان للشخص هاتف واحد فسنضيف سجلا له في جدول الهواتــف.. وإذا كان له أربعة هواتف فسنضيف أربعة سجلات في جدول الهواتف.. وإذا كان له عشرة هواتف فسنضيف عشرة سجلات!.. إذن فليست هناك أي حدود مفروضة علينا هنا، وهذا مناسب لعمل الشركات، حيث يمكن أن يكون للشخص أكثر من هاتف في الشركة، ويحمل أكثر من شريحة محمول، ومتزوج بأكثر من امرأة وبالتالي له أكثر من هاتف منزلي **J**.

والآن دعنا نضيف الخصائص اللازمة إلى الفئة Phone.. مبــدئيا، يجــب أن نتفق على أن كل فئة يجب أن تحتوي على مفتاح أساسي.. وحتى تسهل علــى نفسك، اضغط الخاصية ID في الفئة Person بزر الفأرة الأيمن، ومن القائمــة الموضعية اختر الأمر Copy.. ثم اضغط الفئة Phone بزر الفــأرة الأيمــن، ومن القائمة الموضعية اختر الأمر Paste.. سيؤدي هذا إلى لصق الحقــل ID وإضافته إلى الفئة Phone بنفس اسمه وبكل خصائصه!

والجدول التالي يصف لك الخصائص الأخرى التي يجب أن تضيفها إلى الفئـــة Phone:

Nullable	Type	Source	الخاصية
False	Integer	PersonID	PersonID
False	String	Phone	Phone
True	String	Description	Description

لاحظ أن خانة رقم الصديق وخانة رقم الهاتف لا يمكن تركهما فـــارغتين، وإلا فلن يكون هناك معنى للسجل الذي توجدان فيه.. أما خانة الوصف فيمكن تركها فارغة، أو يمكن أن يكتب فيها المستخدم ملحوظة تخص رقم الهاتف، مثل هــل هو هاتف منزل أم عمل أم هاتف أحد الأقارب أو الجيران... إلخ.

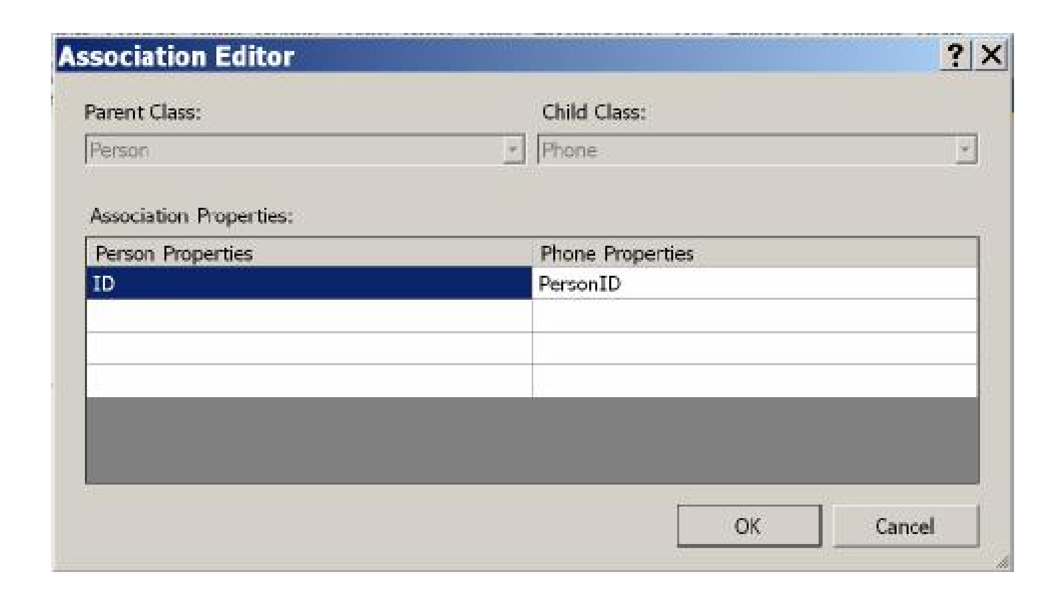
نريد الآن إنشاء علاقة بين جدول الأشخاص وجدول الهواتف.. يمكن فعل هــذا في محتوى البيانات، بإنشاء ارتباط بين الفئة Person والفئـــة Phone.. اتبـــع الخطوات التالية:

١- اضغط العنصر Association في صندوق الأدوات مرة واحـــدة بـــزر الفأرة وارفع إصبعك عن زر الفأرة.

٢- اضغط مخطط الفئة Person مرة واحدة بالفأرة.

٣- اضغط الفئة Phone مرة واحدة بالفأرة.

سيؤدي هذا إلى ظهور نافذة محرر الارتباط، وهي تبدو كما في الصورة:



أعلى النافذة تظهر قائمتان منسدلتان:

- الأولى عنوانها "الفئة الأم" Parent Class.. الفئة الأم هي الفئة الرئيسية في الارتباط، وهي تحتوي على المفتاح الأساسي.. وهنا تم اختيار الفئة Person وتعطيل القائمة لكي لا تغير هذا الاختيار.
- والقائمة الثانية عنوانها "الفئة الابنة" Child Class.. الفئة الابنة هي الفئة الثانوية في الارتباط، وهي تحتوي على المفتاح الفرعي.. وهنا تم اختيار الفئة Phone وتعطيل القائمة لكي لا تغير هذا الاختيار.

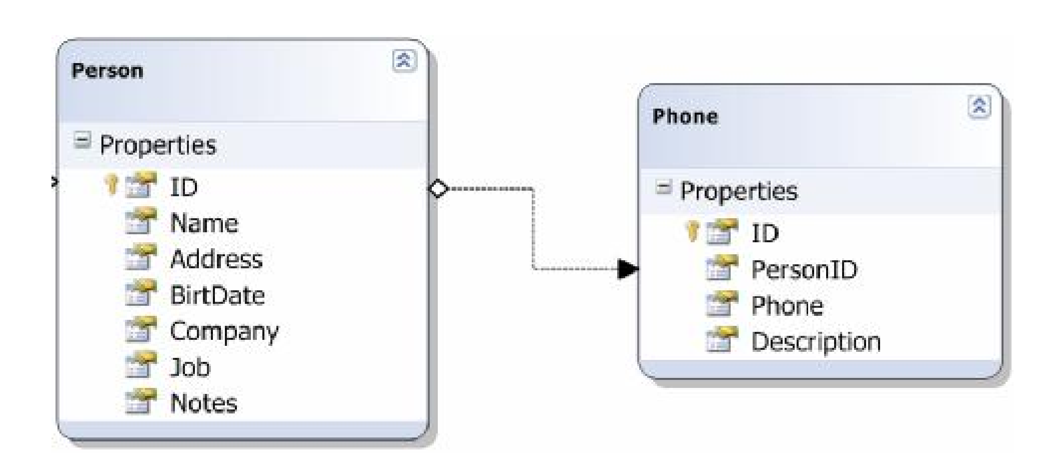
في النصف السفلي من النافذة، يظهر جدول يتكون من عمودين:

- الأول عنوانه "خصائص الشخص" Person Properties، ويجب أن تضيف في هذا العمود الخاصية أو الخصائص التي تعمل كمفتاح أساسي في الارتباط.. وعند الضغط في أي خانة في هذا العمود تظهر قائمة منسدلة فيها خصائص الفئة Person.. اختر منها الخاصية ID.

- والثاني عنوانه "خصائص الهـــاتف" Phone Properties، ويجـــب أن تضيف في هذا العمود الخاصية أو الخصائص التي تعمل كمفتاح فرعي في الارتباط.. وعند الضغط في أي خانة في هذا العمود تظهـــر قائمـــة منسدلة فيها خصائص الفئة Phone.. اختر منها الخاصية PersonID.

بهذا نكون ضبطنا الارتباط ليكون بـــين Person.ID و Phone.PersonID.. اضغط OK لإنشاء هذا الارتباط.. لاحظ أن خطأ سيحدث وتفشل العملية، لو لم يكن لكل من المفتاحين الأساسي والفرعي نفـــس نـــوع البيانـــات (وهـــو هنـــا Integer).

بعد إغلاق النافذة، سترى سهما يصل بين الفئتين Person و Phone ليدل على الارتباط بينهما.



وحينما تضغط هذا السهم بالفأرة سترى نقاطا عليه، يمكنك سحبه منها لتغييـــر شكله وموضعه كما تريد.. ويمكنك أيضا أن تضغطه بزر الفأرة الأيمن ومـــن

القائمة الموضعية تضغط الأمر Edit Association لإعادة عرض نافذة تحرير الارتباط وتغيير اختياراتك.

لاحظ أن إنشاء الارتباط سيضيف خاصيتين إلى الفئتين:

- خاصية اسمها Phones ستضاف إلى الفئة Person، وهي تعيد مجموعة تحتوي على جميع هواتف الشخص الحالي.. أي أن كل عنصر من عناصرها من النوع Phone.. مثلا: إذا كانت النسخة الحالية من الفئة Person تشير إلى شخص اسمه "أحمد" ورقمه ١ في الجدول، فإن الخاصية Phones ستحتوي على جميع هواتف "أحمد"، أو بمعنى آخر: ستحتوي على جميع الصفوف التي يوجد في الخانة PersonID التابعة لها القيمة ١.
- وخاصية اسمها Person ستضاف إلى الفئة Phone، وهي تعيد نسخة من الفئة Person تشير إلى الشخص الذي ينتمي إليه الهاتف الحالي.

لاحظ أن هاتين الخاصيتين لا تظهران على مخطط الفئتين، لكننا نستطيع استخدامهما من الكود.

فئة اللقاء Meeting Class:

أضف فئة جديدة إلى محتوى البيانات اسمها Meeting، واجعل مصدرها في قاعدة البيانات الجدول Meetings (باستخدام الخاصية Source).. هذا الجدول وظيفته أن نضع فيه تفاصيل مواعيد الأشخاص، حيث سنربط الموعد بصاحبه من خلال عمود المفتاح الثانوي PersonID.

إذن مبدئيا، انسخ الحقلين ID و PersonID من الفئة Phone والصقهما في الفئة Meeting.
والجدول التالي يصف لك الخصائص الأخرى التي يجب أن تضيفها إلى الفئة Meeting:

الخاصية	Source	Type	Nullable
Time	Time	Date	False
Duration	Duration	Date	False
Place	Place	String	True
Reason	Reason	String	True
Notes	Notes	String	True
ImportanceID	ImportanceID	Byte	False

لاحظ أننا سنحفظ مدة اللقاء باعتبارها تاريخا.. هذا سيحل لنا العديد من المشاكل عند كتابة الاستعلامات على هذا الجدول كما سنرى لاحقا.. عموما كل ما يهمنا من هذا التاريخ هو جزء الساعات وجزء الدقائق فقط لأنهما يمثلان مدة اللقاء، أما باقي الأجزاء فسنهملها.
بعد هذا أنشئ ارتباطا بين Person.ID و Meeting.PersonID بالطريقة التي تعلمناها.

فئة الأهمية Importance Class:

سيكون من المفيد أيضا أن نعرف فئة نضع فيها وصف أهمية الموعد التي تناظر الرقم الدال على هذه الأهمية.. هذا يتيح بعد هذا إضافة درجات أخرى

من الأهمية، كما أنه سيكون مفيدا لنا في عرض البيانات في جـــدول العـــرض DataGridView كما سنرى لاحقا.

أضف فئة جديدة إلى محتوى البيانات، واجعل اسمها Importance، واجعــل مصدرها في قاعدة البيانــات الجــدول Importances (باســتخدام الخاصــية Source).

أضف إلى هذه الفئة مفتاحا أساسيا اسمه ID وخاصية اســمها Description، واجعل خصائصهما كما هو موضح في الجدول:

Auto Generated Value	Nullable	Type	Source	الخاصية
False	False	Byte	ID	ID
False	False	String	Description	Description

أنشـــئ ارتباطـــا بـــين هـــذه الفئـــة والفئـــة Meeting بـــين الخاصـــيتين Importance.ID و Meeting.ImportanceID.

فئة المستخدم User Class:

أضف فئة جديدة إلى محتوى البيانات، اسمها User، واجعل مصدرها في قاعدة البيانات الجدول Users (باستخدام الخاصية Source).. سنضع في هذا الجدول تفاصيل مستخدمي البرنامج من موظفي الشركة.. ونظرا لأن المستخدم هو في الحقيقة شخص، فسنضع بياناته في الجدول Persons، ونربط بينه وبين جدول المستخدمين عن طريق المفتاح الفرعي PersonID.

إذن مبدئيا، انسخ الحقلين ID و PersonID من الفئة Phone والصقهما في الفئة User.

لاحظ أن كل شخص يمكن أن يظهر في جدول المستخدمين مرة واحدة فقط، لأن جدول المستخدمين هو في الحقيقة امتداد لجدول الشخص في حالة إذا كان موظفا في الشركة ويتاح له استخدام البرنامج.. لهذا تسمى هذه العلاقة علاقة واحد بواحد One-to-One، على عكس علاقة جدول الشخص بجدول الهواتف، فهي علاقة واحد بمتعدد One-To-Many، لأن الشخص الواحد يمكن أن تكون له هواتف متعددة.

وفي الوضع الافتراضي يكون الارتباط الذي ننشئه من النوع One-to-Many، ولتغير هذا في الارتباط بين فئة الشخص وفئة المستخدم، اضغط السهم الممثل للارتباط بزر الفأرة الأيمن، ومن القائمة الموضعية اضغط Properties.. ستجد خاصيتين هامتين في نافذة الخصائص عليك تغيير قيمتيهما، كما هو موضح في الجدول:

الخاصية	القيمة	ملاحظات
الأصل Cardinality	OneToOne	تحدد نوع العلاقة.. نريد هنا علاقة واحد بواحد.
متفرد Unique	True	تجعل المفتاح الفرعي متفردا، لكي لا يظهر نفس رقم الشخص أكثر من مرة مع أكثر من مستخدم.

لاحظ أن إنشاء علاقة من النوع واحد بواحد، يجعل الخاصية التي تضاف إلـــى الفئة الرئيسية مفردة الاسم.. أي أن الفئة Person ستحتوي على خاصية اسمها User تعيد المستخدم الموجود على الطرف الآخر من العلاقة، كمـــا أن الفئـــة User ستحتوي على خاصية اسمها Person تعيد الشـــخص الموجـــود علـــى الطرف الآخر من العلاقة.. والأفضل أن تتأكد أن اسم كل من خاصيتي الـــربط مفرد، وذلك باستخدام الخاصيتين Parent Property و Child Property من نافذة الخصائص، حيث ستجد العلامة + بجوار كل منهما، وعند ضغطها ســـيتم إسدال بعض الخصائص الفرعية، وستجد من بينها الخاصية Name.. تأكد أن هذه الخاصية تحتوي على القيمتين Person و User على الترتيب.

والجدول التالي يصف لك الخصائص الأخرى التي يجب أن تضيفها إلى الفئـــة User:

الخاصية	Source	Type	Nullable
UserName	UserName	String	False
Password	Password	String	False
IsAmdin	IsAmdin	Boolean	False
Enabled	Enabled	Boolean	False

لاحظ أن كل مستخدم سيكون له اسم مستخدم وكلمة مرور ليستخدمهما لتسجيل دخوله إلى البرنامج.. أما الخاصية IsAmdin فهي خاصية منطقية Boolean، لو وضعنا فيها True فسيدل هذا على أن هذا المستخدم أحد مديري النظام، وله صلاحية إضافة وحذف المستخدمين.. أما لو وضعنا فيها False، فسيعني هـــذا

أنه مستخدم عادي يحق له استخدام البرنامج دون أن يستطيع إضافة أو حذف مستخدمين.

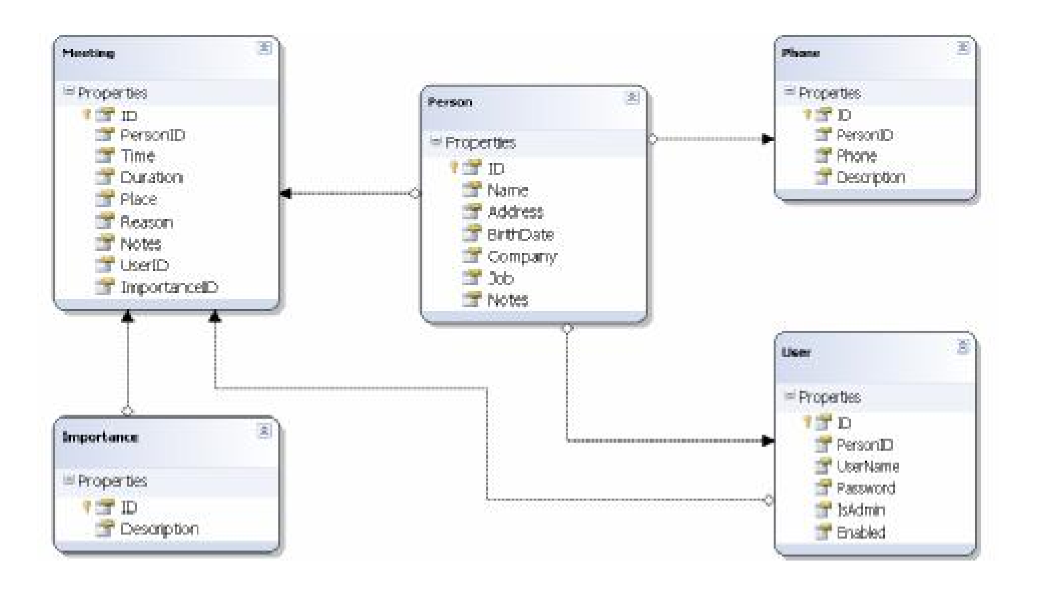
أما الخاصية Enabled فهي تتيح للمدير تعطيل أو تفعيل المستخدم.. فلو وضع فيها المدير القيمة True فهذا يعني أن المستخدم فعال ويمكنه استخدام البرنامج.. أما لو وضع فيها False فهذا يعني أن المستخدم معطل ولا يستطيع استخدام البرنامج.. الحكمة من وراء هذا، هي أن نتحاشى حذف المستخدمين من قاعدة البيانات، لأن حذف المستخدم سيجبرنا على حذف كل مواعيده الموجودة في جدول المواعيد وكذلك حذف بياناته الموجودة في جدول الموظفين.. بينما من الذكاء أن نترك كل البيانات القديمة موجودة كنوع من التسجيل لتعاملات موظفي الشركة، حتى لو تم رفتهم أو منعهم من استخدام البرنامج.. لهذا سنستخدم خاصية التعطيل بدلا من الحذف.

أخيرا: أنشئ ارتباطا بين Person.ID و User.PersonID.

لاحظ أن جدول المواعيد ناقص، فنحن لا نعرف المستخدم الذي سيكون معه الموعد.. لهذا توجه إلى الفئة Meeting وأضف إليها خاصية جديدة اسمها UserID لتعمل كمفتاح فرعي.. استخدم الخاصية Source لتجعل مصدرها العمود UserID، واستخدم الخاصية Type لتجعل نوعها Integer، ثم أنشئ ارتباطا بينها وبين الحقل ID في الفئة User.

ملحوظة:
يمكن أن يحتوي الجدول على أكثر من مفتاح فرعي.. لكنه لا يمكن أن يحتوي إلا على مفتاح رئيسي واحد. ويمكن أن يدخل الجدول في أكثر من علاقة مع عـــدد مـــن الجـــداول الأخرى، سواء تكرر كونه جدولا رئيسيا أو جدولا فرعيا. مثلا: الفئة Person هي الفئة الرئيسية في كل علاقاتهـــا مـــع الفئـــات الأخرى.. والفئة Meeting هي الفئة الفرعية في العلاقتين اللتين دخلت فيهما.

يجب أن يبدو المخطط النهائي لمحتوى البيانات كالتالي:



التحكم في طول الحقول النصية:

في كل الحقول النصية التي أنشأناها في الفئات، لم نحدد أقصى طول مسموح به لكل حقل.. هذا يجعل سيكويل سيرفر يفترض أقصى طول ممكن للحقل.. هذا الطول يكون ٤٠٠٠ حرف إذا كان النص يستخدم الترميز العالمي الموسع Unicode كما في حالة التعامل مع اللغة العربية.. طبعا هذا رقم كبير بالنسبة لاحتياجنا، وعلينا أن نغيره.. لفعل هذا، يمكننا أن نحدد النوع الحقيقي للحقل من بين الأنواع التي تتعامل معها قاعدة البيانات.. بالنسبة للنصوص، سنستخدم النوع nvarchar(L)، حيث:

- الحرف n في الاسم يعني أن النص يستخدم الترميز الموسع.. نحن نحتاج إلى هذا لكي يمكننا حفظ الحروف العربية.. لو استخدمت النوع varchar(L) فستتحول كل الحروف العربية إلى علامات استفهام وستفقد النص الأصلي.
- الحروف var هي اختصار الكلمة variant بمعنى متنوع.. هذا معناه أن النص قد يطول أو يقصر، وسيتم حفظه كما هو بدون إكمال النص القصير بوضع مسافات.. لو استخدمت النوع char فسيتم إكمال النص القصير بوضع مسافات، ليكون لجميع النصوص نفس الطول.
- الحروف char هي اختصار الكلمة Character بمعنى حرف.
- الحرف L بين القوسين يمثل الحد الأقصى لعدد حروف النص.. مثلا: nvarchar(30) يعني أن أقصى نص سيقبله الحقل هو ٣٠ حرفا.

ولكي نحدد نوع البيانات الخاص بسيكويل سيرفر، يمكننا استخدام الخاصية Server Data Type من نافذة الخصائص.. مثلا: عليك أن تضع النص

nvarchar(30) في الخاصية Server Data Type التابعة للحقل Person.Name، لكي تحدد أقصى طول لاسم الشخص بثلاثين حرفا.
استخدم الجدول التالي لتعديل قيم هذه الخاصية في كل الحقول النصية:

Server Data Type	الحقل
nvarchar(30)	Person.Name
nvarchar(500)	Person.Address
nvarchar(50)	Person.Company
nvarchar(20)	Person.Job
nvarchar(500)	Person.Notes
nvarchar(15)	Phone.Phone
nvarchar(20)	Phone.Description
nvarchar(500)	Meeting.Place
nvarchar(500)	Meeting.Reason
nvarchar(500)	Meeting.Notes
nvarchar(10)	User.UserName
nvarchar(10)	User.Password
nvarchar(10)	Importance.Description

إنشاء قاعدة البيانات:

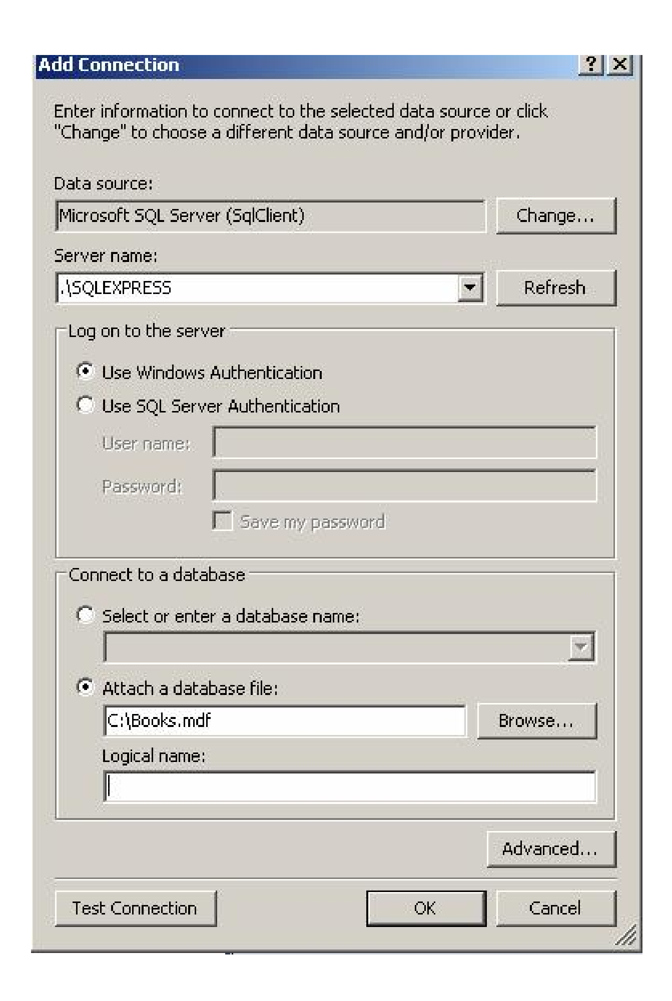
يمكننا أن نطلب من محتوى البيانات، أن يقوم بإنشاء قاعدة البيانات بناء على مخطط الفئات التي أنشأناها.. لكي نفعل هذا يجب أن نضبط خصائص محتوى البيانات أولا.

اضغط بزر الفأرة الأيمن في أي موضع في مصمم الكائنات يخلو من الفئات، ومن القائمة الموضعية اضغط الأمر Properties لفتح نافذة الخصائص.

ستجد في الخاصية Name الاسم SmartNoteBookDBDataContext.. هذا اسم طويل جدا، لهذا غيره إلى SmartNoteBookDB.. لاحظ أننا سنستخدم هذا الاسم في كتابة الكود، كما أنه سيكون اسم قاعـــدة البيانـــات التـــي ســـيتم إنشاؤها.

هناك أيضا خاصية اسمها Context Namespace.. هذه الخاصة تحدد نطاق الاسم الذي سيوضع داخله محتوى البيانات.. ضع في هذه الخاصية الاسم DB.
بالمثل، هناك خاصية اسمها Entity Namespace تحدد نطاق الاســـم الـــذي توضع فيه الفئات التي عرفناها في مصمم الكائنات.. ضع فيها أيضـــا الاســـم DB.

أخيرا، هناك خاصية اسمها Connection، وهي تحدد طريقة الاتصال بخـــادم سيكويل الذي ستعمل عليه قاعدة البيانات.. اضغط زر الإسدال الموجـــود فـــي خانة القيمة، واختر العنصر (New) لإنشاء اتصال جديد.. ستظهر لك النافـــذة الموضحة في الصورة:



في الخانة Server Name اكتب اسم خادم سيكويل الذي ستنشئ عليه قاعـــدة البيانات.. هذا الخادم يمكن أن يكون:

- خادما حقيقيا سواء كان يعمل على شبكة Network أو يعمل عن بعـــد Remote Server، وفي هذه الحالة يجب أن تكتب عنـــوان الإنترنـــت الخاص به مثل 191.0.0.2.. هذا العنوان تحصل عليه من مدير النظام المسئول عن هذا الخادم.

وفي الغالب سيكون هناك اسم مستخدم وكلمة سر، وعليــك أن تختــار اســتخدام تحقيــق هويــة ســيكويل ســيرفر Use SQL Server Authentication لكــي تكتبهمــا فــي الخــانتين User Name و Password.

- خادم محلي Local Server يعمل على جهاز المستخدم.. هذا يسمح لك بتشغيل قاعدة البيانات على نفس الجهاز الذي يعمل عليه البرنامج، وهذا معناه أنها ستكون خاصة بمستخدم واحد فقط.. هذا يحدث أثناء تصميمك للبرنامج لكي يمكنك تجربته على قاعدة بيانات محليــة موجــودة علــى جهازك دون أن تفسد قاعدة البيانات الرئيسية، ويمكن أن يحدث أيضا إذا كنت تريد أن يكون لكل مستخدم قاعدة بيانات مستقلة خاصة به موجودة على جهازه.

ولكي تتعامل مع الخادم المحلي، اكتب الاسم sqlexpress\. في الخانــة Server Name.. هذا هو اسم الخادم المحلي الخاص بسيكويل ســيرفر ٢٠٠٨.. لاحظ أن النقطة في بداية الاسم هي اختصار يرمز إلى اســم جهازك، لأن كل جهاز يعتبر خادما محليا.. لكن نظرا لأن اسم الجهــاز سيختلف من مستخدم إلى آخر، تم الاتفاق على أن يرمز له بالنقطــة أو بالكلمة (local) ويجب أن تضع القوسين حولها.

في هذه الحالة لا تحتاج إلى اسم مستخدم وكلمة سر، لهذا اختر تحقيــق هوية الويندوز Use Windows Authentication.

اضغط الزر Test Connection.. إذا كان الخادم يعمل وكتبت اسمه بشـــكل صحيح، فستظهر رسالة تخبرك أن كل شيء على مـــا يـــرام.. اضـــغط OK لإغلاق النافذة.

الآن يمكننا إنشاء قاعدة البيانات.. اضغط CTRL+S لحفظ التغييـــرات التـــي أجريناها على محتوى البيانات، ثم انتقل إلى النموذج Form1 وضع عليه زرا اسمه BtnCreateDb وضع عليه العنوان "إنشاء قاعدة البيانات".. اكتب فـــي حدث ضغط هذا الزر الكود التالي:

```
Try
    Dim SnbDB As New DB.SmartNoteBookDB
    If Not SnbDB.DatabaseExists Then
       SnbDB.CreateDatabase( )
    End If
Catch Ex as Exception
    MsgBox(Ex.Message)
End Try
```

في بداية هذا الكود، عرفنا متغيرا من نوع محتوى البيانات، لكي يمكننـــا مـــن خلاله إنشاء قاعدة البيانات.. لاحظ أننا كتبنا اسم محتوى البيانات منسوبا إلـــى النطاق DB الذي عرّفناه باستخدام نافذة الخصائص.

وقد استخدمنا وسيلة اسمها DatabaseExists، وهي تعيـــد True إذا كانـــت قاعدة البيانات موجودة بالفعل على سيكويل سيرفر.. طبعا لن نحـــاول إنشـــاء قاعدة البيانات إذا كانت موجودة، حتى لا يحدث خطأ.

بعد هذا استخدمنا الوسيلة CreateDatabase لإنشاء قاعدة البيانات.

لاحظ أن خطأ سيحدث إذا كنت نسيت وضع اسم المصدر Source في أي فئة أو خاصية في محتوى البيانات (ولهذا استخدمنا مقطع معالجة الأخطاء).. فـــي

هذه الحالة سيتم إنشاء قاعدة البيانات، لكن لن يتم إنشاء الجــدول الــذي فيــه المشكلة.. لهذا ربما تريد حذف قاعدة البيانات، لكي تصلح المشكلة، ثــم تعيــد إنشاءها.. لفعل هذا ضع زرا آخر على النموذج اسمه BtnDeleteDb واجعل عنوانه "حذف قاعدة البيانات"، واكتب ما يلي في حدث ضغطه:

```
Try
    Dim SnbDB As New DB.SmartNoteBookDB
    SnbDB.DeleteDatabase( )
Catch Ex As Exception
    MsgBox(Ex.Message)
End Try
```

هذا الكود يعرف متغيرا من فئة محتــوى البيانـــات، ثــم يســتخدم الوســيلة DeleteDatabase لحذف قاعدة البيانات.

والآن، شغل البرنامج، واضغط الزر "إنشاء قاعدة البيانات".. إذا لم يحــدث أي خطأ، فهذا معناه أن قاعدة البيانات قد تم إنشاؤها بنجاح.. في هذه الحالة ســيتم إنشاء ملفين خاصين بها في المجلد:

MSSQL10.SQLEXPRESS\ MSSQL\DATA

على المسار الخاص ببرنامج سيكويل سيرفر، وهو في الغالب:

C:\Program Files\Microsoft SQL Server\

هذان الملفان هما:

- SmartNoteBookDB.mdf: هذا هو ملف قاعدة البيانات.
- SmartNoteBookDB_log.LDF: هذا هو ملف سجل أحداث قاعدة البيانات.

ولا تعمل قاعدة البيانات بدون وجود هذين الملفين معا.

ويمكنك نسخ الملفين ووضعهما في أي مجلد آخر خاص بك كنسخة احتياطيـــة فارغة.

أخيرا، يجب علينا أن نجعل محتوى البيانات يتصل بقاعــدة البيانـــات التـــي أنشأناها.. لاحظ أننا جعلناه يتصل بالخادم فقط، دون تحديد قاعدة بيانات.. نريد الآن أن نجعله يتصل بقاعدة البيانات، لكي يربط الجداول بالفئات التي أنشأناها، مما يمكننا من قراءة البيانات وحفظها في قاعدة البيانات مـــن خـــلال محتـــوى البيانات.

لفعل هذا انقر العنصر SmartNoteBookDB.dbml مرتين فـــي متصـــفح المشاريع لعرض مصمم الكائنات، واعرض خصائص محتوى البيانات.. اضغط الزر الموجود في خانة الخاصية Connection واختر الأمر (New)، وفـــي نافذة إنشاء اتصال اكتب اسم الخادم SqlExpress\.، وفي الجزء السفلي مـــن النافذة فعّل الاختيار:

Select or enter database name

واضغط زر الإسدال الموجود في الخانة الموجودة أسفله.. ستمر لحظات قبل أن تظهر قائمة فيها أسماء قواعد البيانات التي تعمل على الخادم المحلـــي.. اختـــر منها قاعدة البيانات الخاصـــة بنـــا: SmartNoteBookDB واضـــغط OK لإغلاق النافذة.

إنشاء مدير افتراضي للبرنامج:

يجب أن يكون هناك مدير واحد على الأقل في هذا البرنامج، لكي يمكنه إنشـــاء مستخدمين آخرين للبرنامج.. لهذا يجب علينا أن نضيف مديرا افتراضيا، ونحدد

له اسم مستخدم (وليكن Admin) وكلمة سر (ولتكن smartnote) بحيث نعطي هذه المعلومة للمدير الذي سيستخدم البرنامج، ليمكنه تشغيله في أول مرة، وبعد هذا يستطيع تغيير اسم المستخدم وكلمة السر الخاصين به كما يشاء.

أضف زرا جديدا على النموذج Form1 اسمه BtnAddAdmin، واكتب مـــا يلي في حدث ضغطه:

```
Dim SnbDB As New DB.SmartNoteBookDB
Dim Admins = From user In SnbDb.Users
                  Where user.IsAdmin

If Admins.Count = 0 Then
   Dim Pr As New DB.Person With {.Name = "Admin"}
   Dim Admin As New DB.User With {
         .IsAdmin = True, .Enabled = True,
         .UserName = "Admin", .Password = "smartnote",
         .Person = Pr}

   SnbDB.Users.InsertOnSubmit(Admin)
   SnbDB.SubmitChanges( )
End If
```

دعنا نفهم هذا الكود:

في البداية عرّفنا متغيرا اسمه SnbDB لنتعامـــل مـــن خلالـــه مـــع محتـــوى البيانات.. لاحظ أن محتوى البيانات يعرّف خاصية باسم كل جدول في قاعـــدة البيانات.. هذا معناه أن محتوى البيانات الخاص بنا يحتوي علـــى الخصـــائص الموضحة في الجدول:

الخاصية	وظيفتها
Persons	تعيد جدولا صفوفه من نوع الفئة Person.. أي أن هذه الخاصية من النوع Table(Of Person).
Users	تعيد جدولا صفوفه من نوع الفئة User.. أي أن هـذه الخاصية من النوع Table(Of User).
Phones	تعيد جدولا صفوفه من نوع الفئة Phone.. أي أن هذه الخاصية من النوع Table(Of Phone).
Meetings	تعيد جدولا صفوفه من نوع الفئـة Meeting.. أي أن هذه الخاصية من النوع Table(Of Meeting).
Importances	تعيد جدولا صفوفه من نوع الفئـة Importance.. أي أن هذه الخاصية من النوع Table(Of Importance).

ويتحكم محتوى البيانات في كيفية ملء هذه الجداول بالصـفوف مـن قاعـدة البيانات بأفضل كفاءة وسرعة، دون أن نشغل أنفسنا بتفاصيل هذا.
ويسمى كل صف يوضع في هذه الجداول بالكينونة Entity.

الكينونة Entity:
هي نسخة Instance (أو كائن Object) من إحـدى فئـات محتـوى البيانات، تحتوي على بيانات صف من صفوف الجدول.

ويمكنك التعامل مع كل خانة في الكينونة، من خلال الخصائص التي عرّفناها في كل فئة.. فمثلا: يمكن عرض اسم أول شخص في جدول الأشخاص بالجملة التالية:

```
MsgBox(SnbDB.Persons(0).Name)
```

والمريح في الأمر، هو أننا نستطيع استخدام استعلامات LinQ للحصول على البيانات التي نريدها من كل جدول من جداول محتوى البيانات.. وقد استخدمنا استعلاما بسيطا يعيد المستخدمين الذين لهم صلاحيات المدير، وذلك بالتأكد من أن الخاصية IsAdmin الخاصة بهم تحتوي على القيمة True.. بعد هذا سنفحص عدد نتائج هذا الاستعلام، فإن كانت صفرا (أي لا يوجد مديرون)، فسنضيف مديرا افتراضيا إلى الجدول.

لاحظ أننا نستطيع تحسين الاستعلام السابق، فكل ما أردناه منه هو حساب عدد المديرين، فلماذا نرهق قاعدة البيانات بإرسال كل بيانات المديرين إلينا؟.. تذكر دائما أن قاعدة البيانات قد توجد على خادم Server، وأن هذا الخادم عليه ضغوط كثيرة من مستخدمين كثيرين، وأن البيانات التي تطلبها تصل إليك عبر شبكة اتصال، مما يعني أن عملية النقل قد تكون بطيئة، وتزداد بطئا كلما زاد حجم البيانات التي تطلبها.. لكل هذا عليك ألا تطلب من الخادم أية بيانات لا تحتاجها، وأن تكتب أكفأ استعلام يجعلك تؤدي وظيفة برنامجك في أسرع وقت.

إذن، فالأفضل أن نعيد كتابة الاستعلام السابق ليصير كالتالي:

```
Dim AdminsCount = Aggregate User In SnbDB.Users
                  Into Count(User.IsAdmin)
```

هذا الاستعلام يسمى استعلام التجميع Aggregate Query، وهو يبدأ بالكلمة Aggregate بدلا من الكلمة From، ويستخدم الكلمة Into بدلا من الكلمة

Select، وتتبع الكلمة Into إحـــدى دوال التجميـــع (Count، Max، Min، Sum، Avg... إلخ).

ولدالة التجميع Count صيغتان:

- الصيغة الأولى لا تستقبل أي معاملات، وهي تعيد عدد الصفوف الموجودة في الجدول الذي نستعلم عنه.
- والصيغة الثانية تستقبل شرطا، لتعيد عدد صفوف الجدول التي تحقق هــذا الشـــرط.. وقـــد اســـتخدمنا هنـــا الشـــرط User.IsAdmin = True (وكاختصار لم نكتب الجزء True =) لنحصل على عدد الصفوف التـــي تحتوي الخانة IsAdmin فيها على القيمة True.

لاحظ أن القيمة العائدة من استعلام التجميع تكون عددا صحيحا وليس مجموعة نتائج.. لهذا سيكون المتغير AdminsCount من النـــوع Integer، ويمكننـــا استخدامه في جملة الشرط مباشرة كالتالي:

```
If AdminsCount = 0 Then
    ' نفس الكود السابق
End If
```

نريد الآن أن نفهم الكود الذي استخدمناه لإضافة مدير جديد.. مبدئيا يجـــب أن نتفق على أن أي مستخدم يجب أن يكون شخصا، لأن جدول المستخدمين تـــابع لجدول الأشخاص.. لهذا يجب أن نضيف شخصا إلى جدول الأشـــخاص أولا، وليكن اسمه Admin دون الاهتمام بباقي بياناته، حتى يمكننا أن نجعـــل هـــذا الشخص مستخدما في جدول المستخدمين.

لفعل هذا عرّفنا كينونة جديدة من فئة الشخص، بنفس الطريقة التي نعرف بها متغيرا من أي فئة، ووضعنا في الخاصية Name لهذا الشخص الجديد القيمة Admin:

```
Dim Pr As New DB.Person With {.Name = "Admin"}
```

وبالمثل، عرّفنا كينونة جديدة من فئة المستخدم، وجعلنا هذا المستخدم مديرا بوضع القيمة True في الخاصية IsAdmin وجعلناه مستخدما فعالا بوضع القيمة True في الخاصية Enabled، مع تحديد اسم مستخدم وكلمة سر افتراضيين وليكونا (Admin و smartnote).. أخيرا، لكي نربط هذا المستخدم بالشخص الذي يمثله، وضعنا كينونة الشخص في الخاصية Person التي تربط بين الفئتين:

```
Dim Admin As New DB.User With {
            .IsAdmin = True, .Enabled = True,
            .UserName = "Admin",
            .Password = "smartnote",
            .Person = Pr}
```

لاحظ أننا لم نهتم بوضع قيم الخصائص Person.ID و User.ID و User.PersonID، فرقم الشخص ورقم المستخدم سيولدان تلقائيا عند إضافة سجل الشخص وسجل المستخدم إلى قاعدة البيانات، والعلاقة بينهما ستجعل الخاصية User.PersonID تأخذ نفس الرقم الذي سيوضع في الخاصية Person.ID، وذلك لأننا جعلنا المستخدم الجديد يرتبط بالشخص الجديد (.Person = Pr).

لاحظ أن كل ما فعلناه حتى الآن لا يؤثر على جداول محتوى البيانات.. فقد عرفنا كينونتين مستقلتين ولم نضفهما بعد إلى أية جداول.

لإضافة كينونة إلى أي جدول، يمكنك اســتخدام الوســيلة InsertOnSubmit الخاصة بهذا الجدول، ومعناها "الإدراج عند الإرسال".. هــذا يعنــي أن هــذه الوسيلة لن تضيف الكينونة إلى الجدول إلا بعد إرسال التغييرات من محتــوى البيانات إلى قاعدة البيانــات، وهــو مــا يمكــن فعلــه باســتخدام الوســيلة SubmitChanges:

```
SnbDB.Users.InsertOnSubmit(Admin)
SnbDB.SubmitChanges( )
```

وتذكر دائما أن أي تغييرات تجريها على جداول محتــوى البيانــات، لا تــؤثر مباشرة على قاعدة البيانات، بل تظل محفوظة في محتوى البيانات فــي ذاكــرة جهازك، إلى أن تستدعي الوسيلة SubmitChanges.. هذه الوسيلة تقوم بمــا يلي:

١- أي تغيير في بيانات الجداول تحفظه في قاعدة البيانات.
٢- أي سجل أضيف إلى الجداول، تضيفه إلى قاعدة البيانات.
٣- أي سجل حُذف من الجداول، تحذفه من قاعدة البيانات.

لكنّ هناك سؤالا هاما هنا:

لماذا أدرجنا الكينونة Admin في جــدول المســتخدمين باســتخدام الوســيلة InsertOnSubmit، ولم ندرج كينونة الشخص في جدول الأشخاص بالمثل؟

السبب في هذا أننا ربطنا كينونة المستخدم بكينونة الشخص من خلال الخاصية User.Person.. إن محتوى البيانات ذكي جدا، وحينمــا يجــد أن المســتخدم الجديد مرتبط بشخص جديد، فسيضيفهما معا إلى قاعدة البيانات.. لهــذا كــان يمكننا كتابة الكود السابق بالطريقة العكسية، وذلك بإضافة سجل المستخدم إلــى

ســـجل الشـــخص، وطلـــب حفـــظ ســـجل الشـــخص باســـتخدام الوســـيلة InsertOnSubmit:

```
Dim Admin As New DB.User With {
            .IsAdmin = True, .Enabled = True,
            .UserName = "Admin",
            .Password = "smartnote"}
Dim Pr As New DB.Person With {
             .Name = "Admin", .User = Admin}
SnbDB.Persons.InsertOnSubmit(Pr)
SnbDB.SubmitChanges( )
```

وفي كلتا الطريقتين سيتم حفظ سجلي الشخص والمستخدم في قاعدة البيانات.

والآن شغل البرنامج، واضغط زر إضافة المدير، ثم أغلقه.

إضافة درجات الأهمية إلى قاعدة البيانات:

ضع زرا على النموذج Form1 عنوانه "إضافة درجات الأهمية"، واكتب الكود التالي في حدث ضغطه:

```
Dim SnbDB As New DB.SmartNoteBookDB
Dim ImpCount = Aggregate Imp In SnbDB.Importances
                        Into Count( )
If ImpCount = 0 Then
   Dim Imp As New DB.Importance With {
             .ID = 0, .Description = "عادي"}
   SnbDB.Importances.InsertOnSubmit(Imp)
   Imp = New DB.Importance With {
             .ID = 1, .Description = "هام"}
   SnbDB.Importances.InsertOnSubmit(Imp)
```

```
    Imp = New DB.Importance With {
            .ID = 2, .Description = "هام جدا"}
    SnbDB.Importances.InsertOnSubmit(Imp)
    SnbDB.SubmitChanges( )
End If
```

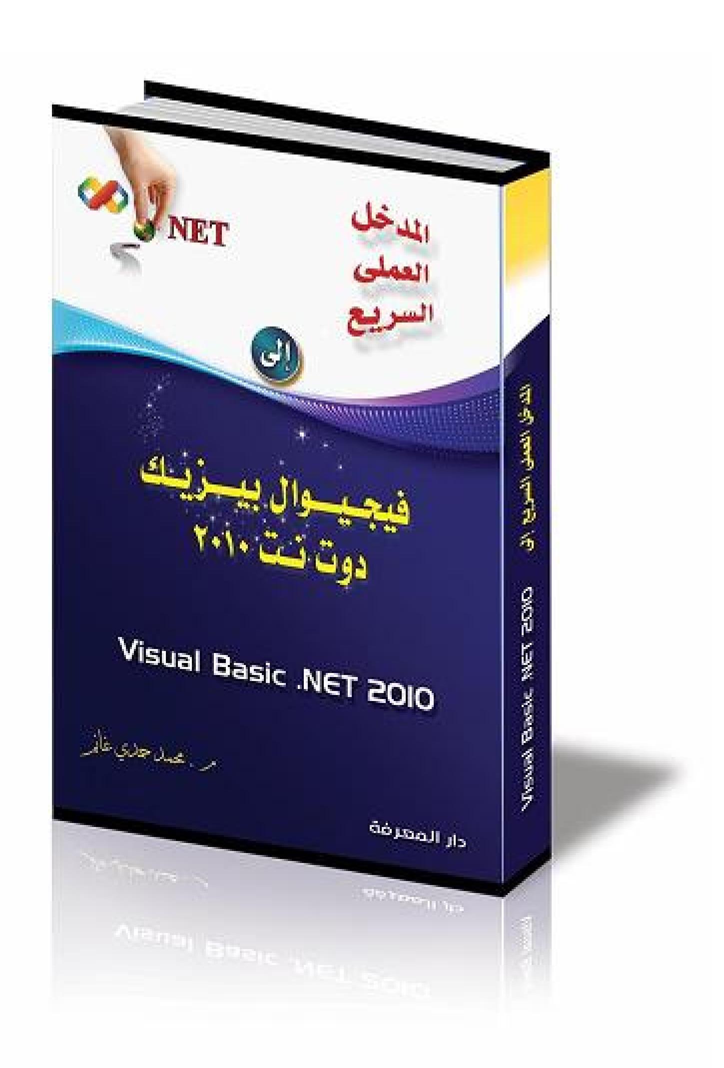
هذا الكود واضح، فهو يستعلم عن عدد السجلات في الجـــدول Importances فإن كان صفرا فإنه يضيف ثلاثة سجلات جديدة إلى هــذا الجـــدول، بتعريـــف كينونة لكل سجل وإضافتها إلى الجدول عند الإرسال، ثم إرسال التغييرات.

والآن شغل البرنامج، واضغط زر إضافة درجات الأهمية، ثم أغلقه.

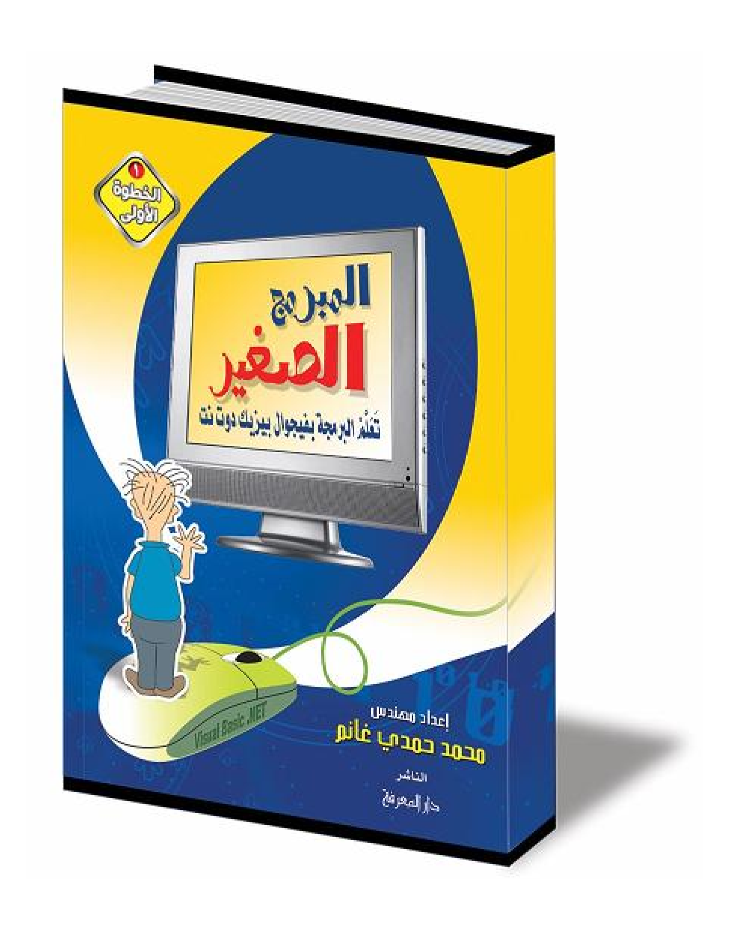
نحن الآن جاهزون لتصميم نماذج المشروع.

ملحوظة:
لن نعرض النموذج Form1 لمستخدم البرنامج ولا حتى للمدير.. هــذا النموذج خاص بنا نحن ــ المبرمجين ــ فقد استخدمناه لإنشاء قاعـــدة البيانات ووضع القيم الابتدائية فيها.

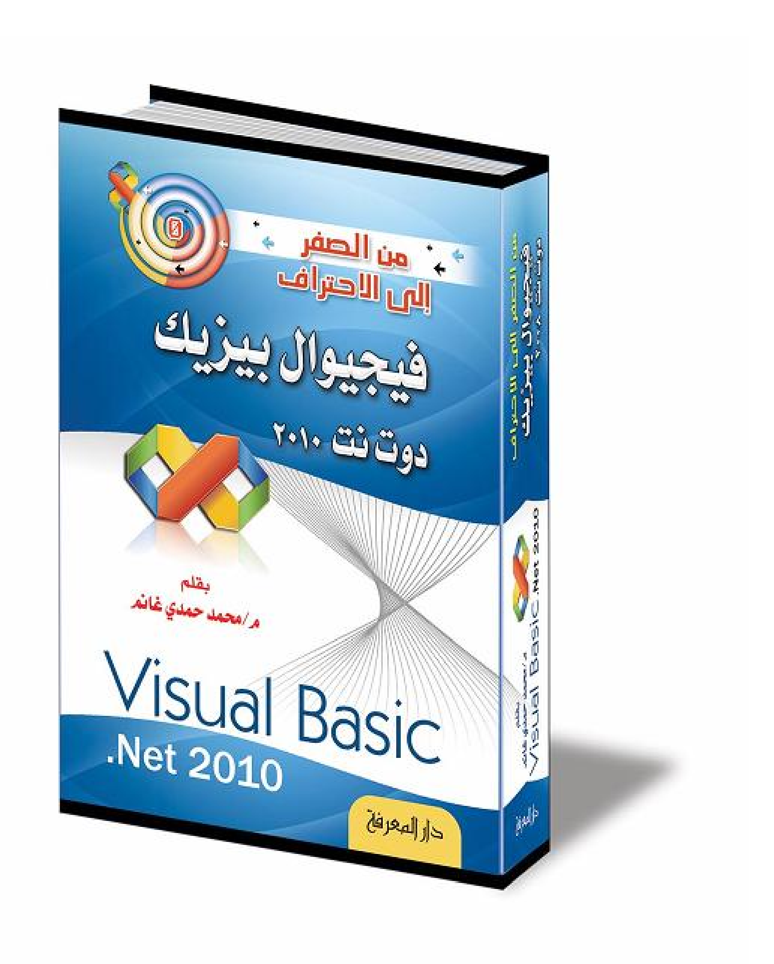
إصدارات م. محمد حمدي غانم



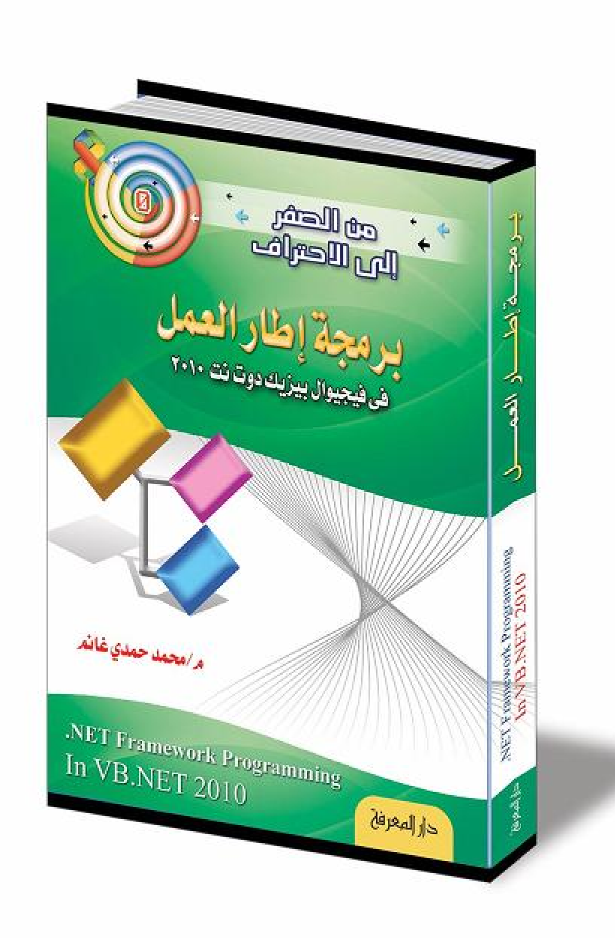
المدخل العملي السريع إلى فيجيوال بيزيك دوت نت ٢٠١٠



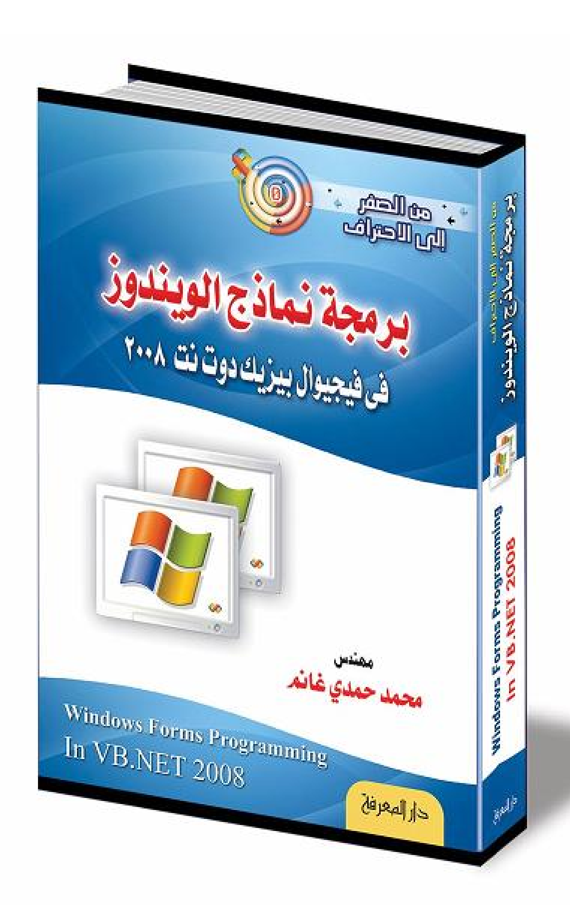
[المبرمج الصغير: تعلم البرمجة بفيجوال بيزيك دوت نت]



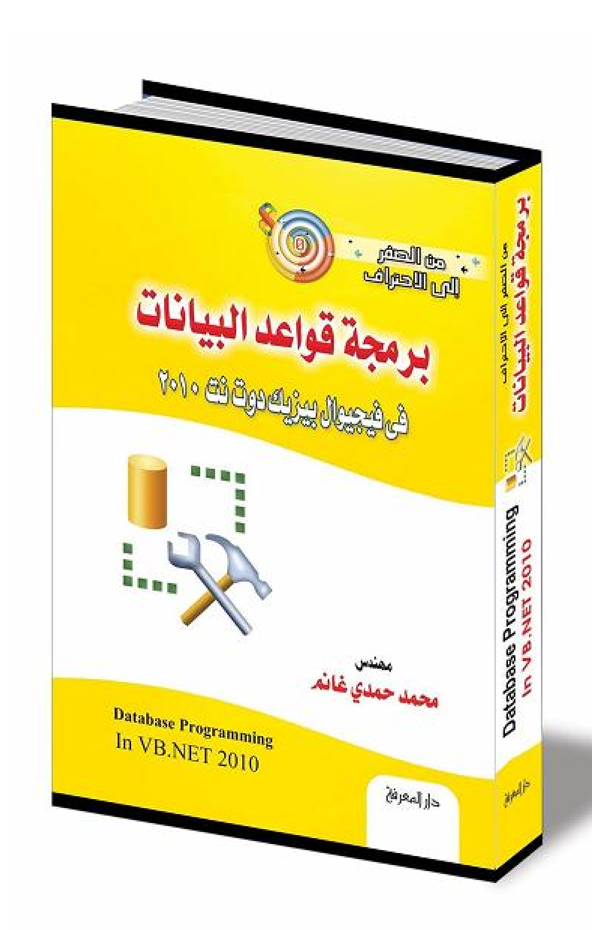
من الصفر إلى الاحتراف: فيجيوال بيزيك دوت نت ٢٠١٠



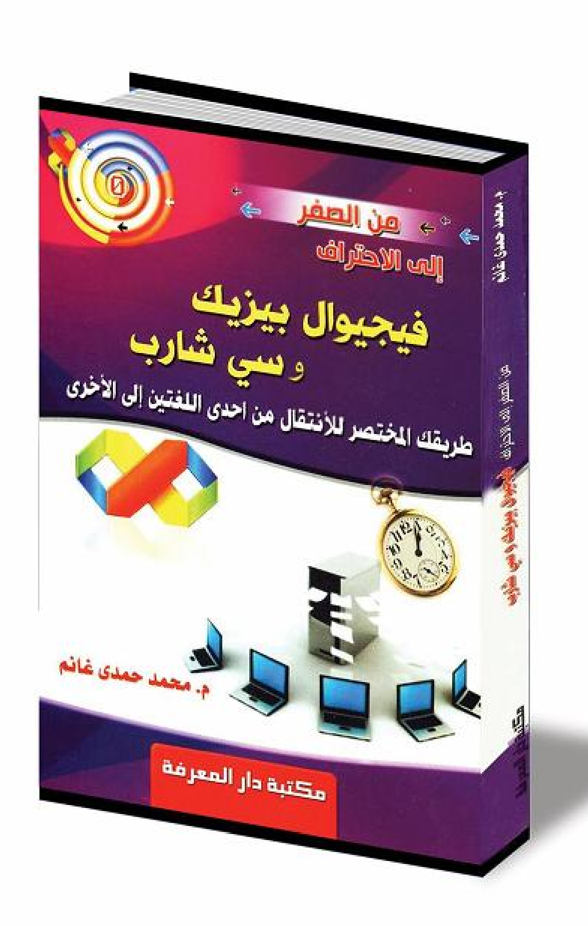
من الصفر إلى الاحتراف: برمجة إطار العمل (VB.NET)



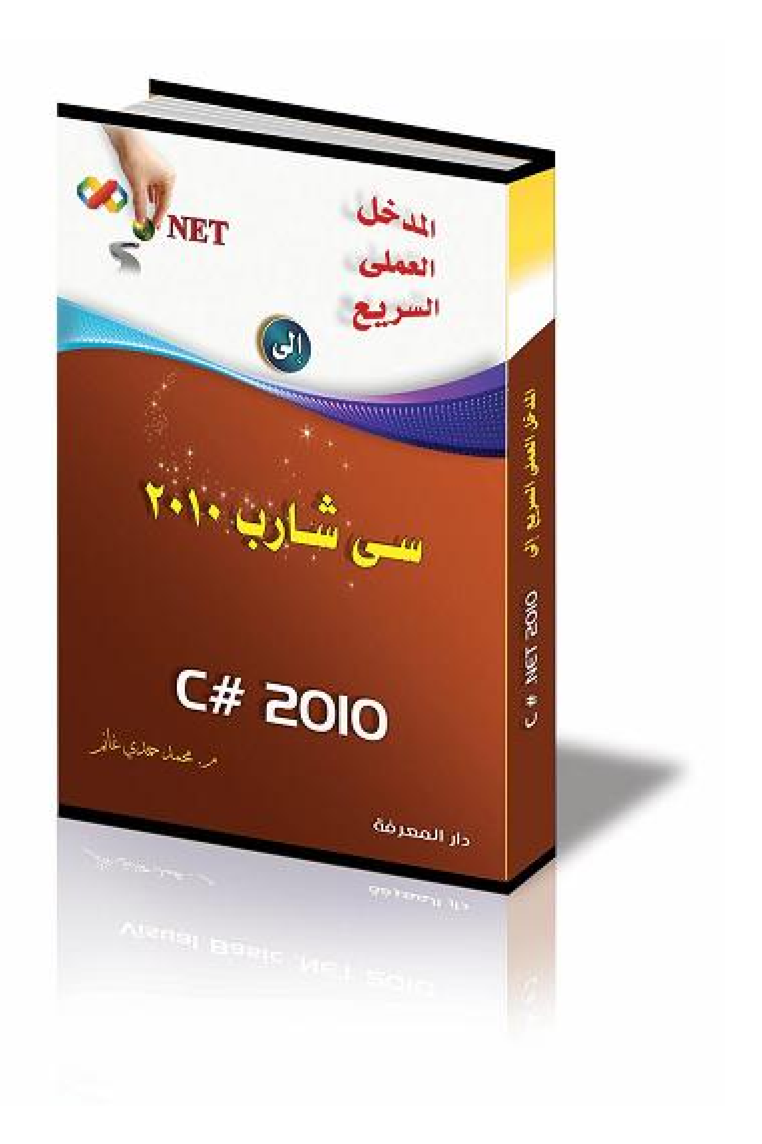
من الصفر إلى الاحتراف: برمجة نماذج الويندوز (VB.NET)



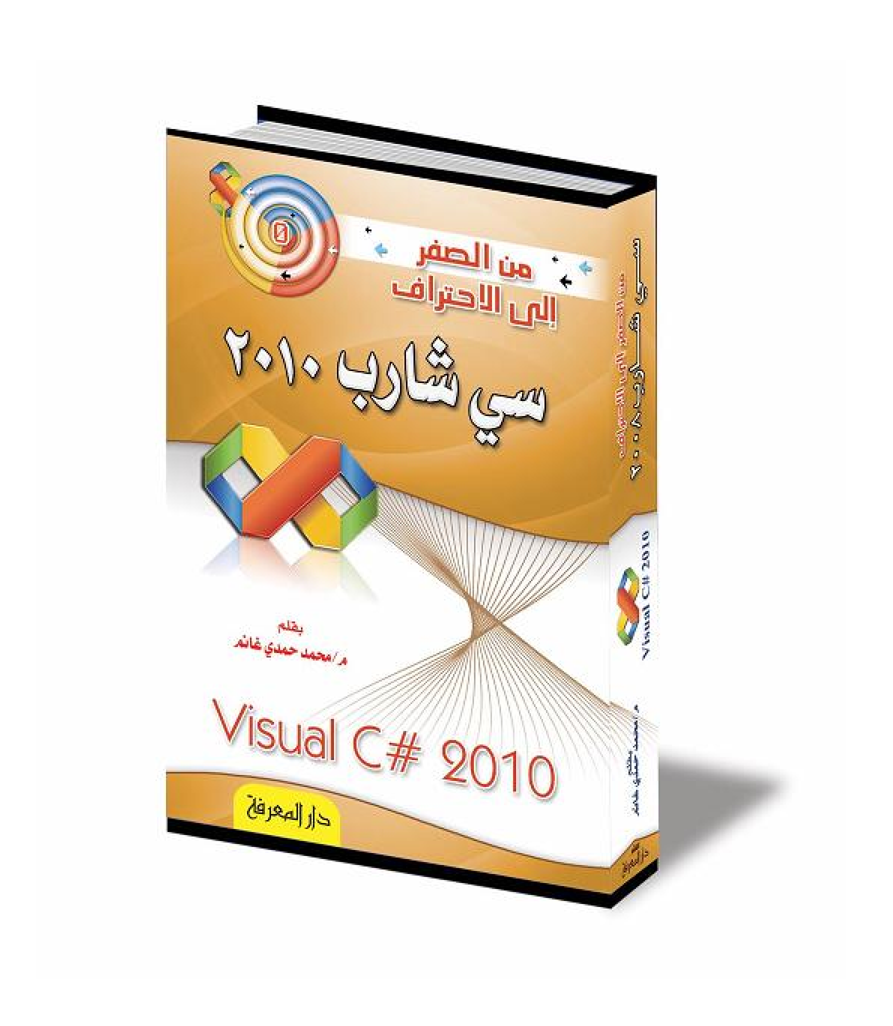
[من الصفر إلى الاحتراف: برمجة قواعد البيانات (VB.NET)]



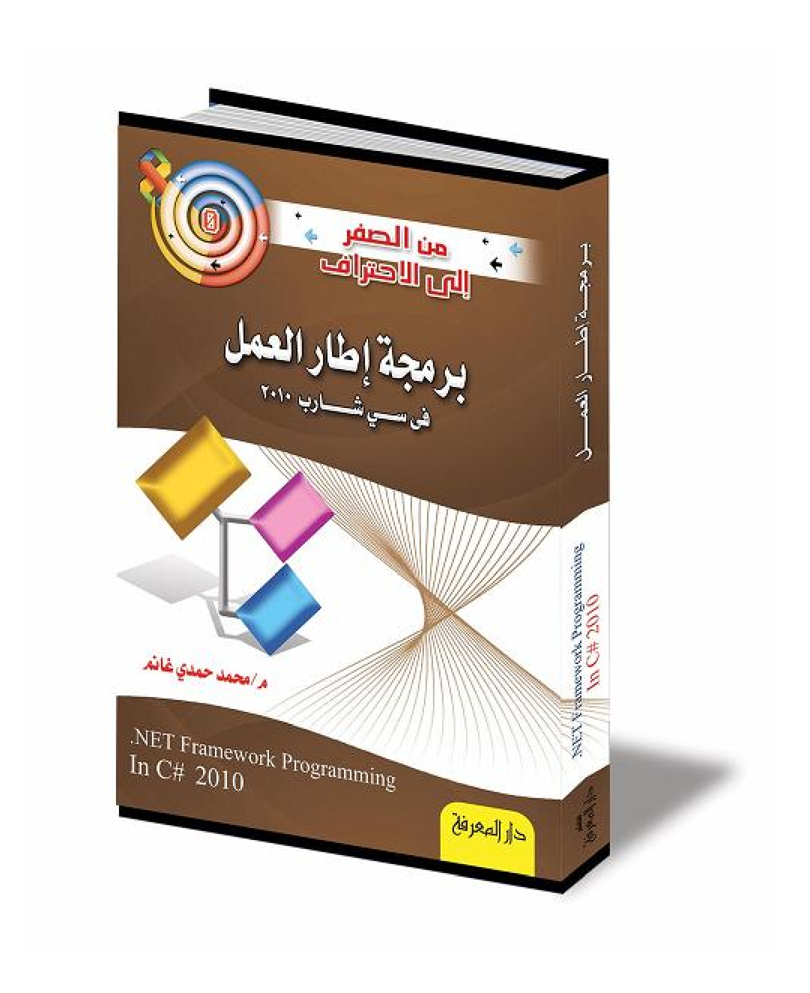
فيجيوال بيزيك وسي شارب: طريقك المختصر للانتقال من إحدى اللغتين إلى الأخرى



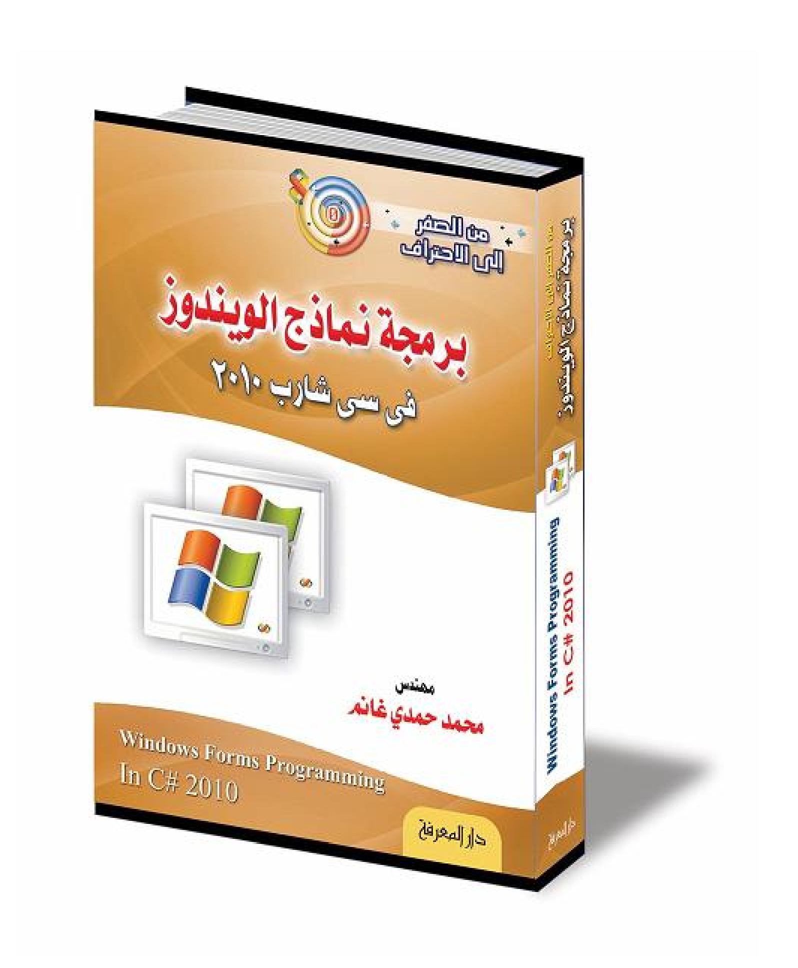
المدخل العملي السريع إلى سي شارب ٢٠١٠



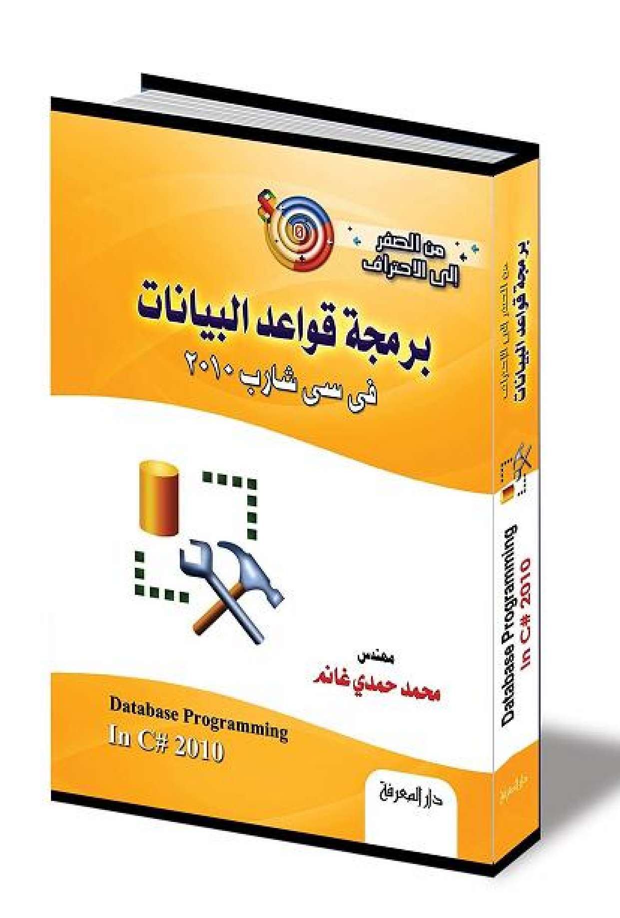
[من الصفر إلى الاحتراف: سي شارب ٢٠٠٨]



من الصفر إلى الاحتراف: برمجة إطار العمل (C#)



من الصفر إلى الاحتراف: برمجة نماذج الويندوز (C#)



[من الصفر إلى الاحتراف: برمجة قواعد البيانات (C#)]



انتهاك حدود اللحظة (ديوان شعر)

أمـــاكـــــن البيـــــع:

- مكتبة دار المعرفة:

٤ شارع السرايات ــ أمام هندسة عين شمس ــ بالقرب من ميدان عبــده باشا ــ العباسية ــ القاهرة.

- دار حراء، دار السحاب، مكتبة شادي، مكتبة النهضة الحديثة:

شارع شريف ــ وسط البلد ــ القاهرة.

- مكتبة الأنجلو:

ميدان باب اللوق ــ القاهرة.

- المكتبة الأكاديمية:

أمام مدخل محطة مترو الدقي، شارع التحرير، الجيزة.

- مكتبة منشأة المعارف ومكتبة علاء الدين:

محطة الرمل ــ الإسكندرية.

- دار المعارف:

فروع: الفجالة، وسط البلد، الإسكندرية، الزقازيق، جامعة الزقازيق، الإسماعيلية، أسيوط.

- للحصول على الكتب في دمياط، يرجى الاتصال بالأستاذ رجب عبد الواحد على الهاتف ٠١٨٦٧١٦٧٩٨ .

- للحصول على الكتب في المنصورة، يرجى الاتصال بالأستاذ أحمد غانم على الهاتف ٠١٩١١٧٦١٣٧ .

- للحصول على الكتب في المنوفية، يرجى الاتصال بالأستاذ عماد عبد الحليم على الهاتف ٠١٠٧٦٧٥٠٩٩ .

- مكتبة العبيكان:

المملكة العربية السعودية.

- **مكتبة الهندسة:**

جازان، المملكة العربية السعودية.

للاستعلام عن أماكن بيع الكتب في باقي المحافظات المصرية، يمكن الاتصال بمكتبة دار المعرفة على الهاتف **٠٢/٢٦٨٤٤٠٤٣** أو عبر البريد الإلكتروني: **dar_elmaarefa@yahoo.com**

للراغبين في إرسال كتبي إليهم خارج مصر عبر البريد:

تكلفة الشحن عبر البريد هي:

٦٠ جنيه (تكاليف ثابتة على الطرد) + ١٠ جنيه على كل كيلو جرام من وزن الطرد.

بالنسبة لكتبي الكبيرة (٧٠٠ و٨٠٠ صفحة)، فمتوسط وزنها حوالي كجم، أي أن إرسال نسخة من أي كتاب منها سيتكلف ١٠ جنيهات مقابل الوزن إضافة إلى ٦٠ جنيه مصاريف الطرد.. أي أن تكلفة الشحن ستكون ٧٠ جنيها مصريا + سعر الكتاب ٧٠ جنيها، أي أن التكلفة الإجمالية ستكون حوالي ١٤٠ جنيها مصريا، أي حوالي ٢٥ دولارا.

بالنسبة لكتاب إطار العمل وكتاب المدخل العمل السريع فكل منهما نصف الوزن ونصف السعر، أي أن طلب نسختين منهما عبر البريد سيتكلف ٢٥ دولارا.

أما الكتابان الصغيران (المبرمج الصغير والانتقال من فيجيوال بيزيك إلى سي شارب)، فهما ربع الوزن وربع السعر، أي أن ٤ نسخ من أي منهما ستتكلف ٢٥ دولارا.

ملحوظة هامة:

كلما زاد عدد نسخ الكتب في الطرد الواحد، قلت التكلفة الإضافية المضافة على سعر كل كتاب، نتيجة لتوزيع تكلفة الشحن على أكثر من كتاب.. كما أن طلب عدد من الكتب يجعل المكتبة تمنحك نسبة خصم، وكلما زاد عدد النسخ تزيد نسبة الخصم.. إن خصما نسبته ١٥% مثلا على ١٠ نسخ من الكتب كبيرة الحجم سيغطي تقريبا تكاليف الشحن.. لهذا سيكون من الأفضل لو اشترك مجموعة من الأصدقاء في طلب مجموعة من الكتب معا كي لا يتحملوا أي تكاليف إضافية على أسعار هذه الكتب.

على الراغبين في الحصول على الكتب بهذه الطريقة، الاتصال بالأستاذ عماد عبد الحليم المسئول في مكتبة دار المعرفة على هاتفه 20107675099+ أو إرسال رسالة SMS مختصرة إليه، بشرط أن يذكر فيها عدد النسخ المطلوبة من كل كتاب، واسم البلد المراد الإرسال إليه، والعنوان، ليحسب تكاليف الشحن، ويخبركم بكيفية التحويل.

مثلا، هذا نص رسالة مقترح:

أريد ٣ نسخ من كتاب برمجة الويندوز (بيزيك)، ونسختين من كتاب قواعد البيانات (سي شارب)..

الاسم: فلان الفلاني

عنوان الشحن: ......

البلد: ....

وسيرد هو برسالة يوضح فيها سعر الشحن، وكيفية تحويله، وعند تلقي النقــود سيرسل الكتب على عنوان الشحن الموضح في الرسالة.

أو يمكنكم مراسلته عبر بريد المكتبة:

dar_elmaarefa@yahoo.com

مع ملاحظة أن رده على البريد متقطع وعلى فترات متباعدة.. رســائل SMS أسرع وأرخص من مكالمات الهاتف، وستحل مشكلة اختلاف اللهجات.

وأعتذر مرة أخرى على المشقة.. لكن ما باليد حيلة.